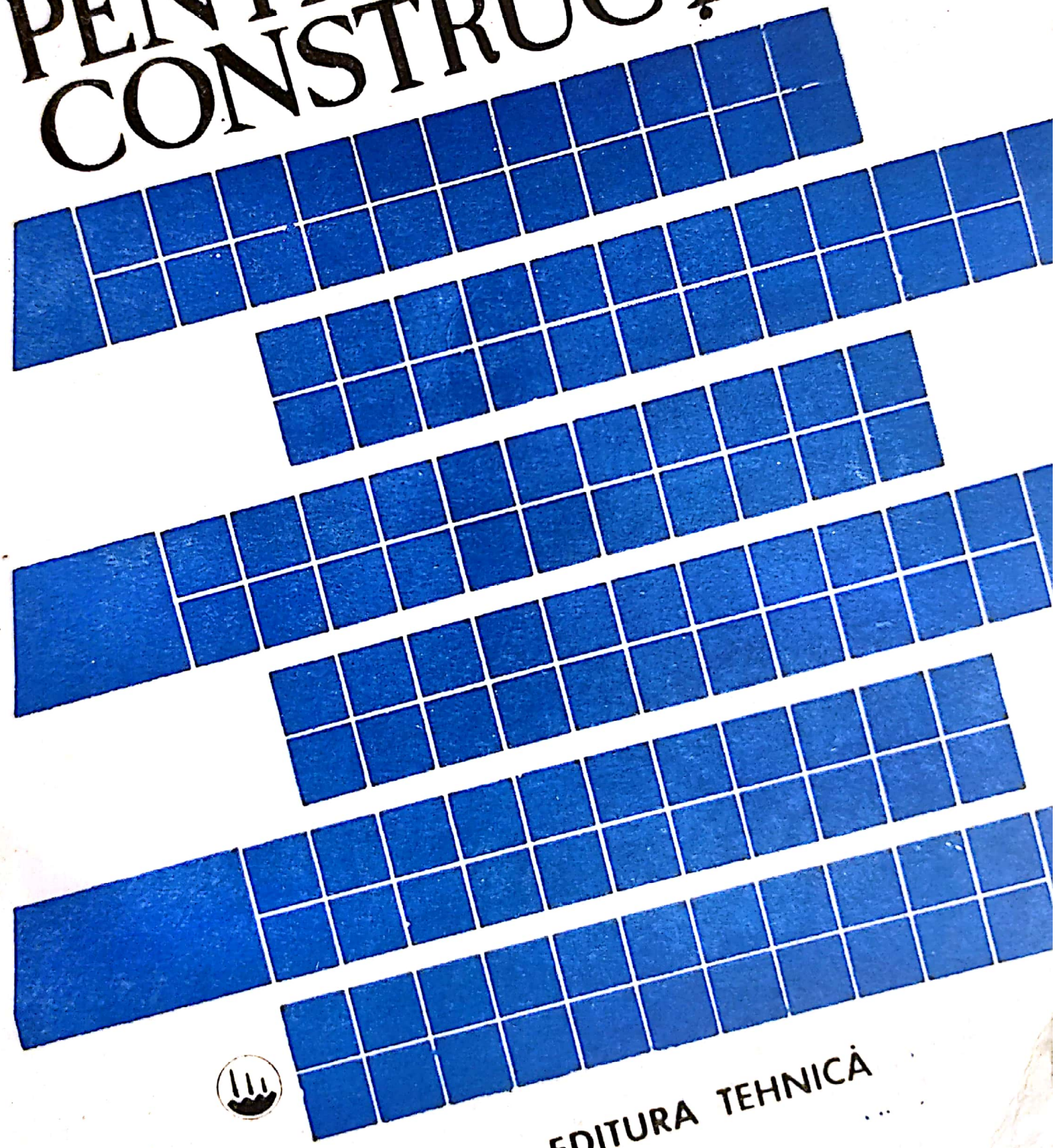
696
523

Ioan Sârbu

CALCULUL INSTALAȚIILOR PENTRU CONSTRUCȚII



EDITURA TEHNICĂ

Scanned with OKEN Scanner

I. Sârbu — NUMERICAL AND OPTIMIZING METHODS IN BUILDING EQUIPMENT DESIGN

The book presents the main numerical and optimizing methods and, even more, the choosen algorithms for computer assistance in order to solve many kind of problems in searching, design and exploitation reffering to the building equipments.
It is structurated in six chapters:

I — Introduction

II — General considerations concerning numerical methods in computing for the building equipments and the processing analyse in the same field

III — Numerical methods in solving of some specific problems in building equipments.

IV — Operational computing methods in optimizing building equipments design — Applications.

V — Optimizing models in equipments computing

VI — Numerical applications and computing programmes.

The book may be used by specialists in all branches (water supply, canalization, gas furnace, heating or cooling instalations etc.), beeing also useful for all kind of engineers, system-analysts, teachers, students, etc.

Scanned with OKEN Scanner

696
S23

dr. ing. IOAN SĂRBU

CALCULUL INSTALAȚIILOR
PENTRU
CONSTRUCȚII

METODE NUMERICE ȘI DE OPTIMIZARE

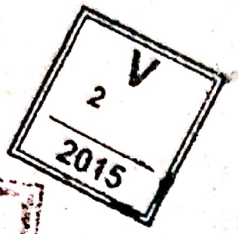
628.959

EDITURA TEHNICĂ
BUCUREȘTI, 1994

Scanned with OKEN Scanner

Copyright © 1994, Editura Tehnică.
Toate drepturile asupra acestei ediții sînt rezervate editurii.

Adresa: EDITURA TEHNICĂ
Piața Presei Libere 1,
33 București, România,
cod 79738



V'23

Control științific: *Prof. dr. ing. CORNEL JURA*
Redactor: *ing. ANTON POPESCU și ADINA NEGOIȚĂ*
Tehnoredactor: *V.E. UNGUREANU*
Coperta: *SIMONA DUMITRESCU*

Bun de tipar: 02.03.1994
Coli tipar: 16,50
C.Z. 696
ISBN 973-31-0525-2

Scanned with OKEN Scanner

Prefață

În domeniul cercetării și proiectării instalațiilor pentru construcții s-au înregistrat în ultimii ani progrese remarcabile. Necesitatea creșterii similitudinii dintre procesele reale din aceste instalații și modelele matematice care le reprezintă a condus inevitabil la sporirea complexității modelelor și implicit la imposibilitatea rezolvării lor fără aplicarea unor tehnici moderne de calcul și utilizarea calculatoarelor electronice numerice.

Elaborarea unei lucrări cu un pronunțat caracter aplicativ privind metodele de calcul numeric și de optimizare programabile pe calculator se înscrie pe linia generală de promovare a calculului automat în diversele domenii ale instalațiilor pentru construcții.

Lucrarea își propune să sistematizeze și să prezinte într-o manieră unitară problemele legate de principalele metode numerice și de optimizare și algoritmii de calcul orientați pe calculator pentru rezolvarea diverselor tipuri de probleme întîlnite în cercetarea, proiectarea și exploatarea instalațiilor pentru construcții, fiind structurată în 6 capitole.

După ce într-un prim capitol introductiv se prezintă unele elemente necesare pe parcursul lucrării privind utilizarea ordinatoarelor electronice în calculele numerice și de optimizare, aspectele matematice ale unui algoritm, tipurile de erori introduse la executarea unui algoritm etc., în capitolul 2 se fac considerații succinte privind metodele de rezolvare a unor probleme de analiză matematică și algebră, aplicabile la calculul numeric al instalațiilor, precum și o prezentare generală a metodelor moderne de modelare și simulare numerică a proceselor din instalații, cu referire îndeosebi la metoda cu diferențe finite și respectiv, metoda cu elemente finite, ultima făcînd parte integrantă din cercetarea și proiectarea asistată de calculator.

Capitolul 3 este destinat mai multor modele de calcul numeric pentru rezolvarea unor probleme specifice instalațiilor, ca: studiul formării gheții și al prevenirii înghețului apei în conductele aeriene sub presiune, analiza rețelelor inelare complexe de distribuție a apei și a energiei termice, analiza numerică a procesului pompării apei în sistemele de alimentare cu apă și căldură și a problemelor de conducție a căldurii, modelarea matematică a proceselor de transfer termic din rezervoarele de acumulare ale instalațiilor de preparare a apei calde, modelarea matematică a procesului de absorbție din instalațiile frigorifice etc.

Capitolul 4 este consacrat tratării unor metode ale calculului operațional, ca programarea matematică (liniară, neliniară, dinamică) și teoria grafurilor, descriindu-se algoritmi aplicabili la optimizarea instalațiilor, de mare actualitate și cu o importanță practică deosebită.

În capitolul 5 se formulează diverse modele complexe de optimizare pentru calculul instalațiilor, cum sînt: calculul hidraulic optim al canalelor cu taluzuri plane și radier cilindric, proiectarea optimală a rețelelor exterioare de distribuție a

Scanned with OKEN Scanner

apei, a gazelor și a căldurii, dimensionarea optimală a rețelelor interioare de alimentare cu apă și căldură în clădiri, dimensionarea optimală a izolațiilor frigorifice, stabilirea încărcării optime a unor sisteme termoenergetice funcționînd în paralel, dimensionarea optimală a unui ansamblu de schimbătoare de căldură etc.

Avînd în vedere scopul aplicativ al lucrării, în ultimul capitol se prezintă o serie de exemple și aplicații numerice pentru o diversitate de probleme reprezentative ce pot fi întîlnite în practică, care facilitează înțelegerea prezentării teoretice, precum și un număr însemnat de programe ordinatoare, elaborate pe baza algoritmilor și modelelor de calcul expuse, în limbajul de programare FORTRAN 77, cu aplicabilitate generală sau pentru rezolvarea unor probleme specializate, ce pot fi rulate pe mai multe calculatoare cu unele modificări neesențiale, dacă ordinatoarele respective dispun de compilatorul asociat limbajului respectiv. Ideea de bază care a stat la concepția programelor a fost aceea de a se constitui într-un instrument de lucru în activitatea de cercetare, proiectare și didactică. Ca urmare, cititorul, folosindu-se și de elementele de programare în limbajul FORTRAN 77 prezentate în acest capitol, cu un minim efort poate înțelege și prelua aceste programe, aplicîndu-le cu maximum de eficiență problemelor practice care îl interesează.

Noțiunile teoretice sînt însoțite în majoritatea cazurilor de scheme logice, precum și de un material grafic adecvat, în felul acesta fiind mult mai accesibile cititorilor. De asemenea, lucrarea este însoțită de o bibliografie vastă pentru ca cititorul interesat să-și poată însuși mai bine partea teoretică.

Prin problemele abordate lucrarea se adresează cu precădere specialiștilor din toate domeniile instalațiilor pentru construcții (alimentare cu apă, canalizare, sanitare, gaze, încălzire, răcire etc.), fiind însă utilă și accesibilă unui grup larg de cititori: ingineri, analiști de sisteme, cadre didactice, doctoranzi și studenți ai institutelor tehnice și toți cei care doresc să-și finalizeze lucrările de cercetare și proiectare cu ajutorul tehnicilor electronice de calcul. Sper ca informațiile oferite de această lucrare să trezească interesul cititorului, determinîndu-l să utilizeze metodele numerice și de optimizare la conceperea modelelor matematice de rezolvare a multor probleme ce apar în practica cercetării, proiectării și exploatării instalațiilor, altele decît cele tratate și să se străduiască să descopere aspecte nesesizate pînă acum în toate domeniile instalațiilor pentru construcții.

Pentru utilele sugestii și minuțioasa recenzie științifică a lucrării de față, prima de acest fel în literatura noastră de specialitate, sînt deosebit de recunoscător apreciatului prof. dr. ing. C. Jura.

AUTORUL

Scanned with OKEN Scanner

Cuprins

1. Introducere 11

1.1. Utilizarea ordinatoarelor în calculele numerice și de optimizare 11
1.2. Aspecte matematice ale unui algoritm de calcul numeric 13
1.3. Instabilitatea numerică a algoritmilor și natura problemelor 14
1.4. Noțiuni generale privind metodele literative 15

2. Considerații generale auspra metodelor numerice de calcul al instalațiilor și de analiză a proceselor din instalații 16

2.1. Rolul meteodelor numerice în calculul instalațiilor 16

2.2. Metodele de rezolvare a unor probleme de analiză matematică și algebră 16
2.2.1. Aproximarea și interpolarea funcțiilor 16
2.2.2. Intgerarea numerică a funcțiilor 18
2.2.3. Rezolvarea numerică a ecuațiilor diferențiale 19
2.2.4. Rezolvarea numerică a ecuațiilor algebrice și transcendente 19
2.2.5. Inversarea numerică a matricelor 20
2.2.6. Rezolvarea sistemelor de ecuații liniare 20
2.2.7. Valori și vectori proprii 21
2.2.8. Rezolvarea sistemelor de ecuații neliniare 22

2.3. Metodele moderne de modelare și simulare numerică 23
2.3.1. Principiul metodei diferențelor finite 23
2.3.2. Principiul metodei elementelor finite 24
2.3.3. Etapele de rezolvare a unei probleme cu metoda elementelor finite 25

3. Modele numerice pentru rezolvarea unor probleme specifice de instalații .. 28

3.1. Studiul formării gheții și al prevenirii înghețării apei în conductele aeriene sub presiune 28
3.1.1. Considerații preliminare 28
3.1.2. Descrierea modelului matematic 29

3.2. Analiza rețelelor inelare de distribuție a apei 32
3.2.1. Considerații preliminare 32
3.2.2. Bazele calculului hidraulic al unei rețele inelare în regim permanent de curgere a apei 33
3.2.3. Descrierea modelului matematic 34
3.2.3.1. Principiul metodei nodurilor 34
3.2.3.2. Formularea și aplicarea algoritmului Newton-Raphson .. 35

3.3. Calculul regimului hidraulic în rețele inelare complexe de distribuție a energiei termice 36
3.3.1. Considerații preliminare 36
3.3.2. Bazele calculului hidraulic al unei rețele termice 37
3.3.3. Descrierea modelului matematic 38

3.4. Analiza numerică a procesului pompării apei în sistemele de alimentare cu apă și căldură 40
3.4.1. Considerații preliminare 40
3.4.2. Descrierea modelului matematic 40

Scanned with OKEN Scanner

3.5. Modelarea matematică a proceselor de transfer termic din rezervoarele de acumulare ale instalațiilor de preparare a apei calde 42
3.5.1. Ipoteze de calcul 42
3.5.2. Modele matematice 42
3.6. Rezolvarea cu metoda diferențleor finite a problemei conducției unidimensionale a căldurii în regim tranzitoriu 45
3.7. Modelarea matematică a procesului de absorbție din instalațiile frigorifice 48
3.8. Analiza numerică cu elemente finite a probemelor de conducție a căldurii 52
3.8.1. Proprietăți caracteristice elementelor finite nodale 52
3.8.1.1. Proprietăți topologice 52
3.8.1.2. Sisteme de coordonare 54
3.8.2. Integrarea numerică a expresiilor elementale 57
3.8.2.1. Aplicarea metodei Newon-Côtes 57
3.8.2.2. Aplicarea metodei Gauss-Legendre 58
3.8.3. Tipuri de elemente finite și funcții de interpolare 59
3.8.3.1. Generarea funcțiilor de interpolare 59
3.8.3.2. Elemente finite unidimensionale 62
3.8.3.3. Elemente finite triunghiulare 62
3.8.3.4. Elemente finite rectangulare serendipiene 63
3.8.3.5. Elemente finite patrulatere cu opt noduri 64
3.8.3.6. Elemente finite curbilinii 65
3.8.4. Modelul analitic al conducției termice 66
3.8.5. Formularea variațională a metodei elementelor finite 68
3.8.6. Modelul numeric cu elemente finite al conducției termice bidimensionale în regim staționar 69
3.8.6.1. Forma generală a ecuației elementelor finite 69
3.8.6.2. Expresia matricei de conductivitate 71
3.8.6.3. Expresia vectorului încărcării termice 73
3.8.7. Modelul numeric cu elemente finite al conducției termice bidimensionale în regim tranzitoriu 74
3.9. Aplicarea metodei reziduurilor ponderea la obținerea ecuațiilor cu elemente finite pentru mișcarea permanentă a unui fluid între doi pereți plani paraleli 76
3.9.1. Metoda reziduurilor ponderea 76
3.9.2. Obținerea ecuațiilor cu elemente finite 78

4. Metode ale calculului operațional aplicabile la optimizarea proiectării instalațiilor 80
4.1. Preliminarii matematice aplicabile la optimizarea proiectării instalațiilor 80
4.1.1. Variabilele de proiectare 81
4.1.2. Restricțiile de proiectare 81
4.1.3. Funcția obiectiv 81
4.1.4. Modelul matematic al unei probleme de optimizare 82
4.2. Elemente de programare matematică 84
4.2.1. Noțiuni generale 84
4.2.2. Optimizarea necondiționată 84
4.2.2.1. Evaluarea analitică a extremului unei funcții fără restricții 84
4.2.2.2. Domenii și funcții convexe 85
4.2.2.3. Metode numrice de evaluare a optimului unei funcții fără restricții 91
4.2.3. Optimizarea condiționată 91
4.2.3.1. Metode analitice de evaluare a extremului unei funcții cu restricții 91
4.2.3.2. Principiul programării liniare 92
4.2.3.3. Formularea generală a programării neliniare 94
4.2.4. Metoda programării dinamice 99
4.3. Elemente de teoria grafurilor 100
4.3.1. Noțiuni generale 100
4.3.2. Algoritmi pentru determinarea unui arbore de valoare minimă .. 102
4.3.2.1. Algoritmul Sollin 102
4.3.2.2. Algoritmul Kruskal 102

Scanned with OKEN Scanner

4.3.3. Algoritmi pentru determinarea unui drum de valoare minimă 102
4.3.3.1. Algoritmul Ford 103
4.3.3.2. Algoritmul Bellman-Kalaba 103
4.3.4. Rețeaua de transport 106
4.3.5. Condiții particulare de aplicare a grafurilor la calculul rețelelor de curgere 106
5. Modele de optimizare pentru calculul instalațiilor 109
5.1. Calculul hidraulic optim al canalelor cu taluzuri plane și radier cilindric 109
5.1.1. Exprimarea elementelor geometrice ale secțiunii transversale .. 109
5.1.2. Model de calcul al profilului optim hidraulic 110
5.2. Optimizarea proiectării rețelelor exterioare de distribuție a apei 112
5.2.1. Algoritim de optimizare a traseului rețelelor ramificate 112
5.2.2. Modele de optimizare pentru echilibrarea hidraulică a rețelelor inelare 115
5.2.2.1. Bazele calculului hidraulic 115
5.2.2.2. Relația funcțională pierdere de sarcină-debit în conductele rețelei 117
5.2.2.3. Modelul „conținutul rețelei" 118
5.2.3. Dimnesionarea optimală a rețelelor inelare complexe 121
5.2.3.1. Bazele calculului optimal de dimnesionare hidraulică 121
5.2.3.2. Criterii de optimizare a dimensionării rețelelor 122
5.2.3.3. Modele de calcul al reparației optime a debitelor de tranzit 124
5.2.3.4. Model neliniar de optimizare a dimensionării rețelelor inelare 126
5.2.3.5. Model liniar de optimizare a dimensionării rețelelor inelare 130
5.3. Optimizarea dimensionării rețelelor interioare de alimentare cu apă în clădiri 134
5.4. Calculul simplificat al rețelelor inelare de distribuție a gazelor bazat pe teoria grafurilor 140
5.5. Optimizarea dimensionării rețelelor de termoficare 143
5.5.1. Principii de funcționare a unui sistem de termoficare 143
5.5.2. Elemente caracteristice calculului hidraulic al rețelelor termice 144
5.5.2.1. Relația funcțională pierdere de presiune-debit în conductele rețelei 145
5.5.2.2. Bazele calculului de dimensionare hidraulică a rețelei 145
5.5.3. Model de optimizare a dimensionării rețelelor termice ramificate bitubulare cu apă 146
5.6. Optimizarea dimensionării izolațiilor frigorifice 148
5.6.1. Considerații preliminare 148
5.6.2. Criteriul economic 150
5.6.3. Criteriul energetic 150
5.6.4. Criteriul economico-energetic 151
5.7. Optimizarea încărcării unor sisteme termoenergetice funcționînd în paralel 155
5.8. Determinarea configurației optime a rețelelor electrice 156
5.9. Optimizarea repartiției sarcinilor termice și electrice pe echipamentele unei centrale electrice de termoficare în cazul unui regim dat 158
5.10. Dimensionarea optimală a unui ansamblu de schimbare de căldură 159
5.11. Determinarea momentului optim de înlocuire a conductelor avariate ale rețelelor de transport 161

Scanned with OKEN Scanner

6. Aplicații numerice și programate de calcul 164
6.1. Considerații preliminare 164
6.2. Aplicații numerice 164
6.3. Elemente de programare în limbajul FORTRAN 205
6.3.1. Considerații generale 205
6.3.2. Structura programelor în FORTRAN 205
6.3.3. Elementele de bază ale limbajului FORTRAN 206
6.3.4. Principalele instrucțiuni ale limbajului FORTRAN 208
6.3.5. Subprograme 213
6.3.5.1. Funcții 213
6.3.5.2. Subprograme externe 214
6.3.5.3. Subprograme BLOCK DATA 215
6.3.6. Lucrul cu fișiere pe disc flexibil în cazul microcalcultoarelor .. 215
6.4. Programe de calcul automat 218
Bibliografie 258

Scanned with OKEN Scanner

1
Introducere

1.1. UTILIZAREA ORDINATOARELOR ÎN CALCULELE NUMERICE ŞI DE OPTIMIZARE

Într-o definire destul de generală, *informatica* poate fi caracterizată ca ştiinţa care se ocupă cu culegerea, codificarea, transmiterea şi mai ales prelucrarea informaţiilor de orice gen, presupunînd, pe de o parte cît mai multă memorie, iar pe de altă parte, calculatoare electronice puternice, dotate cu toată gama de aşa-zise „periferice".

Componentă a creativităţii umane, informatica îşi exercită rolul activ în introducerea progresului tehnic şi modernizarea economiei, calculatoarele electronice fiind de neînlocuit în ştiinţă, tehnologie, economie şi oriunde este necesară prelucrarea unor mari cantităţi de date sau informaţii într-un timp scurt.

Dezvoltarea impetuoasă a tehnicilor de calcul electronic, materializată prin noi generaţii de calculatoare cît şi prin limbaje de programare de nivel tot mai înalt, precum şi interacţiunea unor discipline moderne ca analiza numerică, calculul operaţional şi ingineria programării au condus în ultimii ani la progrese remarcabile în domeniul cercetării şi proiectării moderne a instalaţiilor pentru construcţii.

Pentru a fi în măsură să-şi perfecţioneze continuu programele de calcul şi să le actualizeze în concordanţă cu modificările ce se aduc normelor de calcul şi cu sporirea complexităţii modelelor matematice, inginerul instalator trebuie să cunoască metodele adecvate de analiză şi de optimizare, cît şi un limbaj de programare.

La calculul instalaţiilor, care depinde de mai mulţi parametri, se descrie interdependenţa dintre aceştia cu ajutorul unui model matematic adecvat.

Ordinatoarele electronice sînt sisteme fizice concepute pentru implementarea acestor modele matematice şi pentru rezolvarea lor în mod automat.

Un calculator poate fi programat să rezolve orice problemă corect definită, prin definirea ei înţelegîndu-se conceperea unui *algoritm de calcul* constituit din etape care pot fi codificate cu ajutorul unor secvenţe de instrucţiuni ale calculatorului, ce reprezintă programul ordinator.

Un algoritm constă dintr-o secvenţă finită de operaţii, ordonată şi complet definită, care pornind de la datele de intrare produce rezultatele şi trebuie să aibă următoarele caracteristici esenţiale:

— să fie bine definit, adică operaţiile cerute să se specifice riguros, fără ambiguitate;

Scanned with OKEN Scanner

— să fie descris foarte exact;
— să fie efectiv, adică să se termine întotdeauna după efectuarea unui număr finit de operaţii;
— să fie universal, adică să permită rezolvarea unei clase de probleme.

Un mijloc deosebit de sugestiv pentru descrierea algoritmului de calcul îl constituie *schema logică*, care se realizează grafic din simboluri geometrice cu semnificaţii definite şi prezintă următoarele calităţi:
— permite formarea unei imagini clare şi precise asupra fluxului informaţiei;
— este aproape independentă de un anumit limbaj de programare, putînd fi uşor înţeleasă de toţi;
— nu necesită cunoştinţe avansate şi are o sferă largă de aplicabilitate.

Utilizarea schemelor logice apare îndeosebi la depanarea programelor şi la schimbul de informaţii între diverse grupuri de programatori.

Pentru ca un sistem de calcul să realizeze sarcinile dorite de utilizator, programul introdus în calculator trebuie să poată fi recunoscut de acesta, deci să fie exprimat într-un limbaj cod calculator specific. În scopul înlocuirii muncii dificile de scriere a programelor în cod maşină, au fost create limbajele de programare, care folosesc o scriere simplificată în reprezentarea algoritmilor şi structurilor de date.

Există două niveluri ale limbajelor de programare, primul nivel reprezentîndu-l limbajele de asamblare, iar al doilea limbajele evoluate, denumite limbaje de nivel înalt.

La *limbajele de asamblare*, fiecărei instrucţiuni recunoscute de procesor (cod maşină) îi este ataşat un mnemonic (simbol format din 3 sau 4 caractere literale) ce sugerează natura operaţiei pe care o execută instrucţiunea.

Inconvenientul limbajelor de asamblare este acela că un anumit limbaj are utilitate doar pentru un anumit tip de procesor. Translatarea programelor sursă (în limbaj de asamblare) în programe obiect (în limbaj cod maşină) se face de către un program al sistemului numit asamblor. Cele mai răspîndite limbaje de asamblare sînt ASSIRIS, FAST şi LAF —32.

Limbajele evoluate se apropie mai mult de limbajul natural şi de cel matematic. Translatarea unui program sursă scris în limbaj evoluat se face cu ajutorul unui program compilator. Pentru fiecare instrucţiune din limbajul evoluat compilatorul produce un grup de instrucţiuni în limbajul cod maşină.

De asemenea, translatarea limbajelor evoluate se poate face cu un program interpretor, care traduce cîte o linie — instrucţiune a programului sursă, execută instrucţiunile cod maşină ce rezultă şi apoi trece la instrucţiunea din linia următoare a programului sursă.

Limbajele evoluate sînt mai accesibile utilizatorilor, deoarece sînt mai uşor de învăţat şi de înţeles, oferind totodată posibilitatea unei depanări mai uşoare a programului.

Cele mai răspîndite limbaje evoluate pentru programarea calculatoarelor sînt: FORTRAN, BASIC, PASCAL, TURBO PASCAL, ALGOL, COBOL, dBASE etc.

Un program scris într-un anumit limbaj evoluat poate fi rulat pe un număr mare de calculatoare fără schimbări esenţiale, dacă calculatoarele respective dispun de compilatorul asociat limbajului respectiv.

Orice sistem de calcul poate fi considerat ca fiind format din două componente: *hardware* şi *software*, care cooperează la rezolvarea unei probleme ce poate fi foarte complexă şi laborioasă. Partea de hardware constă din calculatorul propriu-zis şi diverse periferice pentru introducerea datelor şi scrierea rezultatelor. Partea de software constă dintr-un ansamblu de programe

Scanned with OKEN Scanner

cu caracter de bază (program care asigură funcționarea echipamentelor) și cu caracter aplicativ, care reprezintă transpunerea algoritmului problemei de rezolvat într-o formă accesibilă calculatorului.

În rezolvarea unei probleme cu ajutorul calculatorului electronic pot fi evidențiate următoarele etape:

— enunțarea problemei și formularea modelului de calcul, cînd se exprimă matematic relațiile și restricțiile dintre parametrii problemei și se pun în evidență condițiile inițiale, precum și restricțiile referitoare la soluție;

— alegerea metodei numerice de rezolvare a problemei, ce presupune să se țină seama de: precizia impusă, viteza de calcul, necesarul de memorie în funcție de volumul datelor, stabilitatea și convergența metodei etc.;

— descrierea algoritmului metodei numerice pentru care se folosesc schemele logice, ce evidențiază succesiunea logică a etapelor importante din algoritmul de calcul și deciziile logice necesare obținerii soluției;

— elaborarea programului de calcul, ce presupune codificarea cu ajutorul unui limbaj de programare a algoritmului metodei numerice alese și transpuse grafic în schema logică, în vederea executării lui, folosind un calculator electronic;

— testarea rezultatelor, ce presupune aplicarea a diferite procedee de control care să permită verificarea unor rezultate parțiale, detectarea eventualelor erori apărute în calcule și modul de propagare a erorilor, în scopul obținerii de informații necesare opririi sau continuării calculelor.

1.2. ASPECTE MATEMATICE ALE UNUI ALGORITM DE CALCUL NUMERIC

Calculul numeric implică aplicarea matematicii la construcția și algoritmizarea metodelor care pot fi utilizate la obținerea soluției numerice a problemelor cu ajutorul calculatorului electronic.

De foarte multe ori se constată că anumite rezultate ale analizei clasice nu sînt integral folositoare analizei numerice. De asemenea, uneori, chiar dacă soluția analitică a unei probleme date poate fi găsită, aceasta nu poate servi la obținerea soluției numerice, exemplificîndu-se în acest sens următorul sistem de ecuații algebrice:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad (i=1,\ 2,\ \ldots,n) \tag{1.1}$$

Cînd acest sistem are o soluție unică, se poate rezolva analitic cu ajutorul regulii lui Cramer, însă această metodă de rezolvare a sistemelor liniare algebrice este neindicată pentru $n \geqslant 3$ atunci cînd se folosește calculatorul electronic, deoarece trebuie calculați $n+1$ determinanți de ordinul n, iar evaluarea fiecăruia dintre ei implică în general $P_n \cdot n!$ operații de înmulțire, dacă dezvoltarea se face în funcție de minori, unde:

$$P_n = \sum_{j=2}^{n} \frac{1}{(j-1)!}; \qquad \lim_{n\to\infty} P_n = e-1 \tag{1.2}$$

De asemenea, este necesar aproximativ același număr de operații de adunare. În concluzie, rezolvarea sistemului (1.1) prin metoda lui Cramer implică $2P_n(n+1)!$ operații elementare. Dacă $n=20$, numărul operațiilor

Scanned with OKEN Scanner

elementare va fi de aproximativ $16 \cdot 10^{19}$, adică un calculator modern, capabil să execute $2 \cdot 10^6$ operații/s, va trebui să ruleze continuu la această problemă $2 \cdot 10^6$ ani.

În cadrul unei aplicații trebuie să se acorde o atenție egală atît aspectelor matematice, cît și aspectelor legate de calculator, pentru selectarea unui algoritm adecvat rezolvării problemei considerate.

Baza teoretică a algoritmilor o dau metodele numerice, dar eficiența și performanțele lor computaționale țin foarte mult de experiența celui care îi dezvoltă și îi folosește.

Precizia calculelor numerice este parametrul important în alegerea metodelor de calcul. Un algortim de calcul este eficient cînd precizia calculelor este bună. Cu toate performanțele calculatoarelor electronice, precizia rezultatelor este influențată de mai mulți factori.

Soluția depinde de datele inițiale, acestea provenind din măsurări, observații, soluții aproximative ale altor probleme, fapt care face ca la rezolvarea numerică a unei probleme să se introducă erori. Uneori erorile sînt introduse și de modelul matematic cînd acesta nu corespunde în toată intimitatea fenomenului fizic modelat, datorită unor aproximații de modelare. Aceste tipuri de erori se numesc *erori inerente.*

La soluționarea numerică a unei probleme se mai poate produce o *eroare de metodă*, iar în procesul de calcul apar *erori de trunchiere* și *erori de rotunjire.*

În concluzie, *eroarea totală* se compune din cele trei erori menționate: eroarea inerentă, eroarea metodei și eroarea de calcul.

1.3. INSTABILITATEA NUMERICĂ A ALGORITMILOR ȘI NATURA PROBLEMELOR

Dacă o problemă are proprietatea că o mică perturbație în una din date (sau în toate) conduce la mici perturbații în soluția matematică, atunci problema se numește *bine condiționată.*

Dacă mici perturbații, chiar numai într-o parte din datele problemei, conduc la mari perturbații în soluția matematică, atunci problema se numește *slab condiționată.*

Fie D datele exacte care caracterizează o problemă și F funcția matematică ce conduce la obținerea soluției exacte $F(D)$. Aceasta poate fi scrisă sub forma unei aplicații:

$$D \xrightarrow{F} F(D) \tag{1.3}$$

În cazul în care datele sînt afectate de erori sau perturbate, se lucrează cu datele perturbate $D*$ Soluția exactă a problemei cu date perturbate este prezentată sub forma:

$$D* \xrightarrow{F} F(D*) \tag{1.4}$$

Aplicația caracterizată de datele D este bine condiționată dacă $D*$ este apropiat de D și $F(D*)$ este apropiată de $F(D)$ într-un anumit sens, altfel aplicația este slab condiționată.

Pentru a măsura distanța dintre $D*$ și D, precum și distanța dintre $F(D*)$ și $F(D)$, sînt necesare o serie de cunoștințe despre forma datelor și forma soluției. În cazul în care D și $F(D)$ sînt vectori sau matrice, se va examina distanța dintre D și $D*$ respectiv $F(D)$ și $F(D*)$ cu ajutorul normei: $\|D - D*\|$ și $\|F(D) - F(D*)\|$.

Scanned with OKEN Scanner

Fie G un algoritm de calcul pentru rezolvarea unei probleme caracterizate de datele D. Un algoritm G este *stabil* dacă există $D*$ apropiat de D astfel că $F(D*)$ este apropiată de $G(D)$ într-un anumit sens, altfel algoritmul este *instabil*.

Elementul ce caracterizează un algoritm de calcul stabil este faptul că soluția obținută cu ajutorul algoritmului este apropiată într-un anumit sens de soluția exactă a problemei ușor perturbate.

1.4. NOȚIUNI GENERALE PRIVIND METODELE ITERATIVE

Utilizarea calculatoarelor electronice oferă posibilitatea folosirii *metodelor iterative* la rezolvarea anumitor tipuri de probleme. În cadrul unei metode iterative se alege o aproximație inițială (număr sau funcție) și, succesiv, se îmbunătățește această aproximație a soluției (prin iterare) în așa fel ca șirul de soluții îmbunătățite să conveargă către soluția problemei considerate.

Astfel de metode sînt destul de simple pentru a fi utilizate pe un calculator electronic, cu toate că implică destul de multe operații aritmetice, lucru care nu devine un impediment avînd în vedere viteza de calcul pentru calculatoarele din actuala etapă.

În timp ce metodele directe, cel puțin teoretic, converg într-un număr finit de etape, metodele iterative necesită un număr infinit de etape pentru convergență.

La realizarea metodelor iterative trebuie avute în vedere următoarele elemente: metoda să fie convergentă, să se poată determina viteza de convergență și să se stabilească criterii pentru stoparea procesului iterativ în momentul obținerii unei aproximări acceptabile pentru soluția problemei considerate.

Scanned with OKEN Scanner

2

Considerații generale asupra metodelor numerice de calcul al instalațiilor și de analiză a proceselor din instalații

2.1. ROLUL METODELOR NUMERICE ÎN CALCULUL INSTALAȚIILOR

Rolul metodelor numerice în calculul instalațiilor pentru construcții este de a furniza o serie de soluții aproximative pentru acele cazuri în care metodele clasice de analiză analitică și experimentală sînt inoperante sau neeficiente. Totodată, ele permit realizarea unor experimente numerice prin simularea fenomenelor și proceselor hidro și termoenergetice.

Metodele numerice sînt utile și în prelucrarea datelor experimentale sau în determinarea unor expresii de calcul al proprietăților fizice ale agenților termici și frigorifici.

Prin aceasta, metodele numerice contribuie în mod efectiv la deschiderea unor direcții noi, extrem de fertile în cercetarea și proiectarea instalațiilor pentru construcții.

Folosirea metodelor numerice nu constituie un scop în sine. Rezultatul aplicării lor, modelul numeric, urmează a fi implementat pe calculator cu ajutorul unui limbaj adecvat.

Utilizarea eficientă a metodelor numerice presupune existența unor cunoștințe teoretice fundamentale în acest domeniu, precum și a unei experiențe de programare.

2.2. METODELE DE REZOLVARE A UNOR PROBLEME DE ANALIZĂ MATEMATICĂ ȘI ALGEBRĂ

2.2.1. APROXIMAREA ȘI INTERPOLAREA FUNCȚIILOR

În multe aplicații practice apare necesitatea evaluării aproximative a unei funcții $f:[a,b]\to R$, punîndu-se problema determinării unei funcții F, care să aproximeze funcția f, în intervalul $[a,b]$. Se recurge la această aproximare fie atunci cînd nu se cunoaște expresia analitică a lui f, dar se cunosc

Scanned with OKEN Scanner

valorile reale într-un număr finit de puncte ca rezultat al unor măsurări experimentale sau ca soluții ale unei ecuații diferențiale, fie atunci cînd expresia analitică a lui f este destul de complicată și cu ajutorul acesteia calculele sînt dificile.

În cazul aproximării prin interpolare, pentru evaluarea lui $f(x)$ se caută o altă funcție $F(x)$ relativ simplă, astfel ca pentru orice valoare a lui x referită ca nod de interpolare, valoarea lui $F(x)$ să fie suficient de aproape de valoarea lui $f(x)$.

Una din metodele cele mai comune ale interpolării este *metoda grafică*, iar o altă metodă mai precisă de interpolare este *interpolarea polinomială*, caz în care funcția aproximativă se prezintă sub forma unui *polinom algebric*:

$$F(x)=c_0+c_1x+c_2x^2+\ldots+c_nx^n \tag{2.1}$$

iar pentru determinarea coeficienților $c_0, c_1,\ldots,c_n$, potrivit metodei celor mai mici pătrate, se minimalizează distanța dintre f și F:

$$d(f,F)=\sqrt{\sum_{k=0}^{n}[f(x_k)-F(x_k)]^2} \tag{2.2.}$$

în care $x_0, x_1,\ldots,x_n$ sînt $n+1$ puncte cunoscute din intervalul $[a,b]$.

Cunoscînd valorile $f_k=f(x_k)$; $k=1,2,\ldots,n$, aproximarea funcției $f(x)$ poate fi realizată și prin *polinomul de interpolare Lagrange*, de gradul $n-1$:

$$F(x)=\sum_{k=1}^{n}w_kf_k \tag{2.3}$$

în care w_k sînt coeficienții Lagrange, dați de relația generică:

$$w_k=\prod_{\substack{j=1\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j} \tag{2.4}$$

Formula lui Lagrange se poate pune sub forma concentrată dată de către Newton:

$$F(x)=f(x_1)+\sum_{i=1}^{n}[x_1,\ x_2,\ldots,x_{i+1}]\ \ (x-x_1)\ (x-x_2)\ldots(x-x_i) \tag{2.5}$$

în care diferențele devizate de ordinul întîi ale funcției f pe nodurile x_i și x_{i+1} sînt date de relația (2.6), iar diferențele divizate de ordinul $n-1$ se calculează cu relația de recurență (2.7):

$$[x_i,x_{i+1}]=\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} \tag{2.6}$$

$$[x_i,x_{i+1},\ldots,x_{i+n}]=\frac{[x_{i+1},\ldots,x_{i+n}]-[x_i,\ldots,x_{i+n-1}]}{x_{i+n}-x_i} \tag{2.7}$$

$$(n=1,\ 2,\ldots;\ i=1,\ 2\ldots)$$

O altă metodă de aproximare este *interpolarea geometrică*, folosind curba geometrică modificată, cu expresia matematică:

$$F(x)=c_0+cx^{\alpha} \tag{2.8}$$

628.959

BIBLIOTECA JUDEȚEANĂ „GH. ASACHI" IAȘI

Scanned with OKEN Scanner

în care coeficientul c_0 se determină prin extrapolare grafică, iar coeficientul c și exponentul α se calculează aplicînd metoda celor mai mari pătrate, cu ajutorul relațiilor:

$$\lg c = \frac{\sum_{k=1}^{n}\lg(f_k - c_0)\sum_{k=1}^{n}(\lg x_k)^2 - \sum_{k=1}^{n}\lg x_k \lg(f_k - c_0)\sum_{k=1}^{n}\lg x_k}{n\sum_{k=1}^{n}(\lg x_k)^2 - \left(\sum_{k=1}^{n}\lg x_k\right)^2} \tag{2.9}$$

$$\alpha = \frac{n\sum_{k=1}^{n}\lg x_k \lg(f_k - c_0) - \sum_{k=1}^{n}\lg x_k \sum_{k=1}^{n}\lg(f_k - c_0)}{n\sum_{k=1}^{n}(\lg x_k)^2 - \left(\sum_{k=1}^{n}\lg x_k\right)^2} \tag{2.10}$$

Pe baza algoritmilor de interpolare Lagrange și Newton s-au elaborat programele ordinatoare APROL 1 și respectiv APRON, iar algoritmul de interpolare geometrică a fost implementat în cadrul programului ordinator APROGE.

În cazul aplicațiilor practice care necesită pentru aproximarea unor funcții ca funcția de aproximare să coincidă cu funcția de aproximat în punctele de interpolare, precum și ca derivatele funcției de aproximare (de un anumit ordin) să coincidă cu derivatele funcției de aproximat (de un anumit ordin) în anumite puncte din punctele de interpolare date, se folosește *interpolarea hermitiană*, punîndu-se problema găsirii unui polinom unic:

$$F(x) = \sum_{i=1}^{n} c_i (x - \beta)^{m-i} \tag{2.11}$$

în care: $m = n + \sum_{k=1}^{n}\alpha_k$; β — constantă oarecare; α_k — ordinul derivatei în punctul x_k.

O altă metodă de aproximare a funcțiilor este *aproximarea cu funcții spline* [59].

2.2.2. INTEGRAREA NUMERICĂ A FUNCȚIILOR

În multe cazuri practice primitiva $F(x)$ a unei funcții $f(x)$ continuă pe intervalul $[a,b]$ este foarte complicată și nu se poate obține prin procedee elementare, de unde rezultă că și calculul integralei definite după formula Newton-Leibniz este foarte dificil și chiar imposibil.

Metodele de calcul numeric al integralelor constau în înlocuirea unei integrale printr-o combinație, de cele mai multe ori liniară, de valori ale funcției de integrat în puncte din domeniul pe care se face integrarea.

Pentru calculul numeric al integralelor simple există mai multe metode de cvadratură (metoda trapezelor, Simpson, Hermite, Newton-Côtes, Gauss-Legendre etc.) care sînt expuse în diverse lucrări de specialitate [23], [27], [59], [94], [111] și se bazează pe formule de forma:

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n} A_k f(x_k) + R(f) \tag{2.12}$$

în care: $f:[a,b]\to R$ este integrabilă pe $[a,b]$, iar constantele A_k și nodurile $x_k (k=1,\ldots,n)$ sînt astfel alese încît restul $R(f)$ să fie nul cînd funcția f este înlocuită în (2.12) cu un polinom de un anumit grad m.

Scanned with OKEN Scanner

În practică $R(f)$ se neglijează, iar alegerea metodei se face în funcție de forma integrantului, domeniul de integrare și performanțele calculatorului.

Pe baza formulelor de integrare numerică după metodele trapezului, Simpson și Newton-Côtes s-au elaborat respectiv programele ordinatoare TRAPEZ, SIMPSON și NEWCOT.

2.2.3. REZOLVAREA NUMERICĂ A ECUAȚIILOR DIFERENȚIALE

Modelele multor procese fizice se exprimă prin ecuații diferențiale de forma:

$$y' = f(x, y) \tag{2.13}$$

cu condiția inițială:

$$y(x_1) = y_1 \tag{2.14}$$

Rezolvarea cu ajutorul calculatoarelor numerice a acestor modele se realizează prin discretizare, obținîndu-se astfel o soluție aproximativă.

Metodele cu diferențe, de tip Euler [59], [94], [111], sînt caracterizate printr-o simplitate a formulelor de iterare, deci facilitate în programare, dar în general și de erori mari.

Metodele Runge-Kutta [12], [94], [111], inițiate de Runge, ca o generalizare a formulelor de cvadratură Simpson și perfecționate ulterior de Kutta, Gill, Merson și alții, păstrează suplețea metodelor clasice, în condițiile creșterii substanțiale a preciziei și a stabilității în calcul. Utilizarea unei formule Runge-Kutta poate permite un pas de integrare destul de mare făcînd ca timpul de calcul să fie optim și dă posibilitatea schimbării pasului de integrare în cursul iterației.

Pe baza algoritmilor Euler și Runge-Kutta de rezolvare numerică a ecuațiilor diferențiale s-au elaborat programele ordinatoare EULER și RUNKUT.

2.2.4. REZOLVAREA NUMERICĂ A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE

La calculul valorii unui polinom cu coeficienți reali:

$$P(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n \tag{2.15}$$

pentru $x = \xi$, se poate aplica schema lui Hörner.

Pentru calculul valorilor derivatelor succesive ale lui P se deduc formulele:

$$P(\xi) = A_n, \quad P^i(\xi) = i!\, A_{n-i} \quad (i = 1, 2, \ldots, n) \tag{2.16}$$

în care:

$$A_{n-i} = P_i(\xi), \tag{2.17}$$

valorile polinoamelor $P_i(\xi)$ determinîndu-se utilizînd repetat schema lui Hörner:

$+a_0$	a_1	a_2	$a_3 \ldots a_{n-1}$	a_n	$\vert\xi$
	$b_0\xi$	$b_1\xi$	$b_2\xi \ldots b_{n-2}\xi$	$b_{n-1}\xi$	
b_0	b_1	b_2	$b_3 \ldots b_{n-1}$	$b_n = P(\xi)$	
	$c_0\xi$	$c_1\xi$	$c_2\xi \ldots c_{n-2}\xi$		
c_0	c_1	c_2	$c_3 \ldots c_{n-1} = P_1(\xi)$		

Un număr considerabil de modele matematice asociate fenomenelor fizice din instalații se reduc în final la o ecuație algebrică sau transcendentă de forma:

$$f(x)=0 \tag{2.18}$$

în care funcția $f:[a,b]\to R$ este derivabilă pe intervalul $[a,b]$.

Atunci cînd expresia lui f este complicată, metodele de rezolvare analitică a ecuației (2.18) devin imposibile, recurgîndu-se la o serie de metode de calcul numeric (metoda bisecției, metoda secantei, metoda Newton, metoda Cebîșev, etc.), descrise în mai multe lucrări de specialitate [26], [62], [67], [94], [111].

Algoritmii de rezolvare a unei ecuații algebrice ($f(x)$ este un polinom) cu metodele Newton și Cebîșev s-au implementat în cadrul programelor ordinatoare ECNEW și ECCEB.

2.2.5. INVERSAREA NUMERICĂ A MATRICELOR

În unele domenii ale instalațiilor pentru construcții apar probleme liniare sau neliniare și probleme de valori și vectori proprii, care implică în rezolvarea lor calcule numerice matriceale.

Matricele inverse pot fi definite numai în cazul matricelor pătratice nesingulare ($|A|\neq 0$), prin relația:

$$A^{-1}=\frac{A^*}{|A|} \tag{2.19}$$

în care A^* este matricea adjunctă matricei A, iar $|A|$ — determinantul matricei A.

Inversarea matricelor poate fi realizată pe lîngă relația de bază (2.19) și prin unele metode numerice:

— metoda partiționării (Frobemus-Schur);
— metoda eliminării (Gauss);
— metoda Cholesky,

care sînt prezentate în lucrări de specialitate ca [23], [94], [111].

Metoda partiționării este o combinație între relația de definire a inversei unei matrice și scrierea partiționată a ei, iar metoda eliminării se bazează pe eliminarea gaussiană.

Metoda Cholesky, foarte laborioasă și ingenioasă în același timp, reține în memoria calculatorului numai elementele nenule, fiind esențială în inversarea matricelor mari de tip bandă.

Algoritmul de inversare a matricelor pătratice prin metoda partiționării și prin metoda eliminării s-a implementat în cadrul programului ordinator INVMAP și respectiv INVMAE.

2.2.6. REZOLVAREA SISTEMELOR DE ECUAȚII LINIARE

Mai multe probleme din domeniul instalațiilor pentru construcții necesită pentru soluționare rezolvarea unor sisteme de ecuații algebrice liniare. De asemenea, programarea liniară, aplicată la soluționarea unor probleme de optimizare a instalațiilor, implică rezolvarea de sisteme de ecuații liniare, ce se scriu sub una din formele:

$$AX=B \tag{2.20}$$

$$\sum_{j=1}^{n} a_{ij}x_j=b_i \qquad (i=1,2,\ldots,n) \tag{2.21}$$

Scanned with OKEN Scanner

în care: A este matricea sistemului, B — vectorul termenilor liberi; X — vectorul necunoscutelor.

Pentru rezolvarea numerică a sistemelor liniare s-au conturat două tipuri de metode:

— *metode directe* sau *exacte* (Gauss, Gauss-Jordan, Cholesky), în care o secvență de operații se execută o singură dată, iar rezultatele obținute sînt o aproximație a rezultatului exact;

— *metode indirecte* sau *iterative* (Jacobi, Gauss-Seidel), care permit găsirea soluției printr-un proces de aproximări succesive. O aceeași secvență de operații este repetată de mai multe ori, obținîndu-se o soluție din ce în ce mai bună în sensul preciziei.

Aceste metode sînt prezentate în numeroase lucrări de specialitate [12], [23], [26], [27], [59], [111].

Se alege acea metodă care este convenabilă din punctul de vedere al unei viteze sporite de convergență pentru o aproximare inițială adecvată.

Metodele indirecte sînt iterative în sensul că se trunchiază șirul infinit de operații în momentul în care este atinsă precizia dorită, pe cînd metodele directe sînt finite deoarece implică un număr finit de operații și bine determinat de dimensiunea sistemului și de metoda aleasă.

Metodele directe de rezolvare a sistemelor algebrice liniare au avantajul că necesită un număr fix de operații elementare pentru un număr de ecuații dat și au dezavantajul că acumulează erori de rotunjire, care conduc la o eroare relativă destul de mare în soluție.

Metodele iterative au dezavantajul că apare posibilitatea de neconvergență și implică un număr mare de operații elementare chiar atunci cînd converg destul de rapid.

Pe baza algoritmilor Gauss și Cholesky s-au elaborat respectiv programele ordinatoare SILGAU și SILCOL, iar algoritmii iterativi Jacobi (iterații simultane) și Gauss-Seidel (iterații succesive) s-au implementat în cadrul programelor ordinatoare SILJAC și SILGAS, care pot servi la rezolvarea numerică a sistemelor de ecuații liniare.

2.2.7. VALORI ȘI VECTORI PROPRII

Problema valorilor și vectorilor proprii apare într-o varietate de aplicații, ca de exemplu în hidraulică la calculul regimurilor de curgere laminară, iar în electricitate la stabilirea caracteristicilor de rezonanță.

Pentru o matrice pătratică $A=[a_{ij}]_{n,n}$, parametrul scalar λ se numește *valoarea proprie*, iar vectorul $X\neq 0$ se numește *vector propriu* corespunzător valorii proprii λ dacă:

$$AX=\lambda X \tag{2.22}$$

Pentru determinarea vectorilor proprii $X^k\neq 0$ care să satisfacă ecuația matriceală (2.22) pusă sub forma dezvoltată:

$$\begin{bmatrix} a_{11}-\lambda_k & a_{12} & \dots a_{1n} \\ a_{21} & a_{22}-\lambda_k \dots a_{2n} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1} & a_{n2} & \dots a_{nn}-\lambda_k \end{bmatrix} \begin{Bmatrix} x_1^k \\ x_2^k \\ \vdots \\ x_n^k \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \tag{2.23}$$

Scanned with OKEN Scanner

trebuie determinate rădăcinile $\lambda_k (k=1,\ldots,n)$ ale ecuației caracteristice:

$$\lambda^n + c_1\lambda^{n-1} + \ldots + c_{n-1}\lambda + c_n = 0 \tag{2.24}$$

iar apoi se rezolvă sistemul liniar omogen (2.23) cu una din metodele convenabile amintite anterior.

Pentru determinarea valorilor proprii ale matricei A este necesar să se construiască polinomul caracteristic și să se rezolve ecuația caracteristică.

Metodele speciale ce se utilizează pentru calculul valorilor proprii se pot grupa în:

— metode care determină direct polinomul caracteristic, prin coeficienți (Krîlov, Leverrier);

— metode care determină valorile proprii și vectorii proprii prin procedee iterative fără a calcula polinomul caracteristic (Jacobi).

Aceste metode sînt descrise în lucrări de specialitate ca [23], [59], [94] [111].

Metoda Leverrier este aplicabilă în cazul în care A este o matrice rară ($a_{ij}=0$ pentru aproape toți i,j).

Pe baza algoritmului Krîlov și Jacobi s-a elaborat programul ordinator PCKRIL și respectiv VAVEPJ.

2.2.8. REZOLVAREA SISTEMELOR DE ECUAȚII NELINIARE

Unele probleme din diverse domenii ale instalațiilor pentru construcții conduc direct la sisteme de ecuații neliniare, care scrise vectorial au forma:

$$F(X) = \{f_1(x_1,x_2,\ldots,x_n)\ldots f_n(x_1,x_2,\ldots,x_n)\} = 0 \tag{2.25}$$

unde ansamblul necunoscutelor $x_1,x_2,\ldots,x_n$ și al funcțiilor $f_1,f_2,\ldots,f_n$ formează cîte un vector:

$$X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}; \qquad F = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{Bmatrix} \tag{2.26}$$

Pentru rezolvarea sistemului de ecuații neliniare (2.25) se pot utiliza metodele iterative Newton-Raphson, suprarelaxărilor succesive, gradientului [23], [62], [94], [111] pe baza primilor doi algoritmi elaborîndu-se programele ordinatoare SINNER și SINSUR.

Folosirea algoritmului Newton-Raphson prezintă inconvenientul că verificarea condițiilor teoremei de existență și unicitate a soluției este laborioasă, neputîndu-se programa și conducînd la riscul ca soluția să depindă de aproximația inițială.

Din punct de vedere al programării la calculator metoda gradientului este destul de avantajoasă. De exemplu, față de metoda Newton-Raphson se înlocuiește rezolvarea unui sistem liniar la fiecare pas de iterație cu înmulțirea unei matrice cu un vector, calculul devenind mai simplu. Are însă dezavantajul că asigură numai o convergență lentă.

Scanned with OKEN Scanner

2.3. METODELE MODERNE DE MODELARE ŞI SIMULARE NUMERICĂ

2.3.1. PRINCIPIUL METODEI DIFERENŢELOR FINITE

În ştiinţă şi tehnologie, unul dintre principalele modele de cunoaştere a mediilor continue este cel de aproximare prin discretizare, constînd în descompunerea domeniului analizat într-un număr finit de elemente discrete şi aproximarea întregului prin ansamblul elementelor componente. S-au dezvoltat două tehnici numerice moderne de aproximare a mediilor continue prin discretizare cum sînt metoda diferenţelor finite şi metoda elementelor finite, care au aplicabilitate şi în rezolvarea unor probleme ce apar la calculul hidraulic şi termic al instalaţiilor pentru construcţii.

Fenomenele fizice sînt descrise din punct de vedere matematic de ecuaţii diferenţiale, prin a căror integrare, în condiţii la limită date, se obţine soluţia exactă a problemei. În acest fel se poate calcula valoarea funcţiei sau a funcţiilor în orice punct al domeniului studiat. Aceasta este calea analitică, obişnuită de rezolvare, aplicabilă însă numai în cazul problemelor relativ simple. Problemele care intervin în activitatea practică sînt de cele mai multe ori complexe, încît integrarea ecuaţiilor diferenţiale este dificilă sau imposibilă. Rezolvarea analitică nu se poate efectua decît prin crearea unui model într-atît de simplificat încît integrarea ecuaţiilor diferenţiale să fie realizabilă. Se obţine astfel soluţia exactă, dar pentru un model simplificat. Deseori este însă preferabil ca, în locul soluţiei exacte a unui model simplificat, să se dispună de o soluţie aproximativă a problemei reale.

Soluţiile aproximative, obţinute prin metode numerice, sînt susceptibile să reflecte, în multe cazuri, mai bine realitatea decît soluţiile exacte ale unor modele simplificate, dar necesită un efort de calcul mai mare, inconvenient ce se elimină în condiţiile existenţei calculatoarelor electronice.

O asemenea metodă numerică este *metoda diferenţelor finite*, concepută sub forma calculului manual încă de pe vremea lui Euler, în cadrul căreia ecuaţiile diferenţiale sînt înlocuite prin ecuaţii algebrice cu diferenţe finite. Soluţia problemei se obţine într-un număr finit de puncte ale unei reţele rectangulare, printr-o aproximare de natură matematică.

Dezavantajul principal al metodei cu diferenţe finite îl constituie folosirea unei reţele rectangulare de discretizare a domeniului de analiză. Deci, folosirea ei pe domenii cu contururi sau suprafeţe curbe introduce o serie de dificultăţi şi artificii de calcul.

De aceea, această metodă are un domeniu mai restrîns de aplicabilitate (probleme uni şi bidimensionale) faţă de metoda elementelor finite. Caracteristica comună a celor două metode o constituie dependenţa convergenţei către valorile exacte, de numărul de elemente în care se discretizează domeniul de analiză sau, cu alte cuvinte, de fineţea de discretizare.

Este evident că folosirea acestor metode numerice este de neconceput fără apelarea la ordinatoarele electronice.

Existenţa continuităţii condiţiilor la limită sau a unor discontinuităţi a acestora doar pe un număr redus de intervale este o restricţie importantă în aplicarea metodei diferenţelor finite la analiza unor medii continue.

Scanned with OKEN Scanner

2.3.2. PRINCIPIUL METODEI ELEMENTELOR FINITE

Aproximarea prin discretizare constituie un concept fundamental, față de care metoda cu elemente finite reprezintă un caz particular.

Metoda elementelor finite este un procedeu de rezolvare aproximativă, cu ajutorul calculatorului electronic, a unei varietăți largi de probleme inginerești. Este o metodă modernă de calcul numeric aplicată pentru prima dată în anul 1950 la calculul mecanic al construcțiilor metalice aeronautice. Primele publicații referitoare la domeniul calculului structurilor datează din 1954, aparținînd lui Turner, Clough, Martin și Topp. Ulterior, Zienkiewicz aplică această metodă în domeniul curgerii fluidelor și electromagnetismului, pentru ca Oden, în anul 1972 să o extindă într-o serie de probleme ale continuumului material.

În aceste probleme se urmărește determinarea, într-un domeniu considerat, a valorilor uneia sau mai multor funcții necunoscute cum sînt de exemplu temperaturile, presiunile, vitezele, deplasările, tensiunile mecanice etc., după natura problemei tratate. Cum domeniul are o infinitate de puncte, valorile funcției sînt în număr infinit.

Metoda elementelor finite folosește ca punct de plecare un model integral al fenomenului studiat, care poate fi obținut în mod direct prin calcul sau poate fi derivat din modelul diferențial corespunzător cu ajutorul calculului variațional sau al metodei reziduurilor ponderate. Potrivit calculului variațional, rezolvarea unei ecuații diferențiale într-un anumit domeniu și cu anumite condiții la limită este echivalentă cu minimalizarea în acel domeniu a unei mărimi funcționale corespunzătoare ecuației diferențiale și condițiilor la limită date.

Spre deosebire de metoda diferențelor finite această metodă se bazează pe aproximarea locală a variabilelor de cîmp pe porțiuni sau subdomenii ale domeniului analizat, denumite *elemente finite*, legate între ele în puncte numite *noduri*. Ca urmare a minimizării funcționalei în toate elementele finite în care a fost împărțit domeniul de analiză și asamblării pe întreg domeniul a efectelor obținute pe elementele finite, rezultă un sistem de ecuații algebrice prin a cărui rezolvare se determină valorile funcției studiate în noduri. În scopul minimizării funcționalei pe elementele finite ale domeniului analizat, funcția sau funcțiile necunoscute, continue pe tot domeniul, sînt aproximate printr-un set de funcții convenționale, continue numai pe cuprinsul elementelor finite. Se spune că aproximarea oferită de metoda elementelor finite este de natură fizică.

În inginerie, aplicațiile metodei elementelor finite pot fi grupate în trei clase de probleme:

a) *probleme de echilibru sau de regim staționar*, în care funcția sau funcțiile necunoscute nu depind de timp. Aici se încadrează problemele de transfer termic în regim staționar, studiul regimurilor staționare de curgere a lichidelor, studiul comportării elastice a corpurilor în regim static etc.;

b) *probleme de valori proprii*, în care parametrii sînt de asemenea independenți de timp și în care se determină anumite valori critice ale parametrilor în condițiile respectării configurației de echilibru. În hidraulică asemenea probleme sînt calculul regimurilor de curgere laminară, în electricitate stabilirea caracteristicilor de rezonanță etc.;

c) *probleme de propagare sau de regim tranzitoriu*, în care funcțiile necunoscute sînt dependente de timp. Transferul căldurii în regim tranzitoriu, regimurile de curgere nestaționară a lichidelor, studiul dinamic al comportării elastice și neelastice a structurilor sînt exemple de asemenea probleme.

Scanned with OKEN Scanner

Principalele avantaje ale metodei elementelor finite față de metoda diferențelor finite sînt următoarele:

— posibilitatea de a modela forme neregulate prin folosirea de elemente finite cu forme și dimensiuni diferite, adecvate configurației domeniului studiat;

— posibilitatea de a trata fără dificultăți probleme în care proprietățile fizice ale mediului variază, cum sînt mediile neomogene, anizotrope, stratificate etc.;

— posibilitatea considerării oricăror condiții la limită pentru problema studiată;

— posibilitatea elaborării unor algoritmi și programe cu grad mare de generalitate, capabile să rezolve o gamă largă de probleme dintr-un anumit domeniu de specialitate.

Metoda elementelor finite are și unele dezavantaje, ca:

— datele de intrare sînt în general numeroase, astfel că este necesar un efort relativ mare pentru pregătirea și introducerea lor;

— rezultatele numerice se obțin sub forma unor ample colecții de valori numerice ale funcției sau funcțiilor studiate, într-un număr relativ mare de noduri;

— calitatea rezultatelor depinde de experiența și abilitatea programatorului de a elabora un model cu elemente finite pentru problema studiată.

2.3.3. ETAPELE DE REZOLVARE A UNEI PROBLEME CU METODA ELEMENTELOR FINITE

Procesul de lucru caracteristic metodei elementelor finite este constituit din două etape principale: *etapa modelării* și *etapa aplicațiilor numerice*. Scopul primei etape îl constituie realizarea unui model analog cu elemente finite pentru modelul analitic care descrie fenomenul fizic cercetat. Implementarea acestui model pe calculator se face sub forma unui program ordinator. Scopul celei de a doua etape este de a folosi acest program de calcul la soluționarea unor probleme concrete, iar rezultatul ei îl constituie soluția numerică a problemei.

Etapa modelării reprezintă un proces specific fiecărui domeniu de investigare, în sensul că în cadrul ei se operează cu ecuațiile diferențiale care descriu o anumită clasă de probleme fizice.

În etapa aplicațiilor numerice se presupun cunoscute modelul analitic și modelul cu elemente finite ale problemei, iar operațiile efectuate pot fi grupate convențional în mai multe faze:

a) **alegerea tipului de elemente finite** și respectiv a funcțiilor de interpolare se face în funcție de variația parametrilor analizați (de exemplu, o variație liniară, parabolică etc.), de geometria domeniului investigat și de capacitatea disponibilă de calcul.

Din punct de vedere al geometriei lor, elementele finite pot fi de tip uni, bi sau tridimensional reprezentînd bare, suprafețe (plane sau curbe) și volume. Fiecare dintre acestea au ca noduri obligatorii punctele extreme și uneori puncte auxiliare situate pe laturi sau interioare. În fiecare nod pot exista una sau mai multe funcții necunoscute, denumite grade de libertate, care se aproximează prin funcțiile de interpolare, iar numărul acestora pentru întregul element finit se obține înmulțind cu numărul de noduri ale elementului. Laturile elementelor finite pot fi drepte sau curbe.

Scanned with OKEN Scanner

De exemplu, o problemă plană poate fi rezolvată folosind triunghiuri, dreptunghiuri sau elemente izoparametrice. Dintre acestea, elementele triunghiulare şi mai ales izoparametrice au o sferă mai largă de aplicabilitate, datorită adaptării lor la domenii limitate de contururi cu multe frînturi. La rîndul lor, aceste elemente prezintă diverse tipuri cu un număr diferit de grade de libertate.

Dacă se dispune de un program de calcul, această fază se concretizează prin selectarea unei opţiuni privind folosirea bibliotecii de elemente finite.

b) **discretizarea domeniului de analiză** înseamnă generarea reţelei de elemente finite şi a matricei de conexiuni. Această fază poate fi realizată manual sau automat de către calculator printr-o subrutină specializată a programului ordinator. Rezultatul acestei operaţii constă dintr-un set de elemente finite, un set de noduri şi o matrice de conexiuni între acestea. Totodată se calculează coordonatele globale ale nodurilor.

c) **constituirea ecuaţiilor elementelor finite şi evaluarea matricelor elementale.** Comportarea mediului în cuprinsul unui element finit este descrisă de ecuaţiile elementale finite denumite şi ecuaţii elementale. Acestea alcătuiesc un sistem de ecuaţii al elementului. Numărul de ecuaţii ale sistemului este egal cu numărul gradelor de libertate pe element.

Ecuaţiile elementelor pot fi deduse pe mai multe căi: metoda variaţională, metoda reziduală (Galerkin), metoda bilanţului energetic.

Cea mai folosită este metoda variaţională, al cărui principiu a fost menţionat la punctul 2.3.2.

Metoda reziduurilor se foloseşte în special cînd metoda bazată pe calculul variaţional nu se poate aplica, de exemplu din cauză că nu se cunoaşte o mărime funcţională pentru problema respectivă.

Cu excepţia folosirii unor elemente simple, evaluarea matricelor elementale se face cu ajutorul metodelor de integrare numerică.

Ca urmare a aplicării uneia din aceste metode se obţine un sistem de ecuaţii de forma:

$$[k]\{\Phi\}_e = \{p\} \tag{2.27}$$

în care: $\{\Phi\}_e$ este vectorul funcţiilor nodale necunoscute pe elementul finit; $[k]$ — matrice a caracteristicilor fizico-geometrice ale elementului finit, denumită matrice de rigiditate (conductivitate); $\{p\}$ — vectorul încărcării (mecanice, termice etc) pe elementul finit. Ultimele două terminologii au fost preluate din mecanica structurilor.

d) **asamblarea ecuaţiilor elementale în sistemul de ecuaţii al domeniului discretizat** impune ca, în nodurile comune elementelor, funcţia sau funcţiile necunoscute să aibă aceeaşi valoare. Asamblarea ecuaţiilor constă în asamblarea matricelor de rigiditate $[k]$ ale elementelor finite în matricea de rigiditate a sistemului şi a vectorilor încărcării pe elemente $\{p\}$ în vectorul încărcării pe întreg domeniul.

Matricea de rigiditate prezintă inconvenientele structurale de a avea dimensiuni mari şi a conţine foarte mulţi coeficienţi nuli, necesitînd un spaţiu apreciabil de memorie. Această situaţie a condus la dezvoltarea unor mijloace specifice de optimizare cum sînt tehnicile de structurare a coeficienţilor nenuli în benzi şi de minimizare a lăţimii lor, precum şi utilizarea metodei frontale de rezolvare progresivă a sistemului de ecuaţii pe măsura asamblării elementelor.

Scanned with OKEN Scanner

e) **rezolvarea sistemului de ecuații.** Metodele numerice de rezolvare a sistemelor de ecuații cu elemente finite se pot grupa în metode liniare și metode neliniare. Cele mai folosite metode de rezolvare a sistemelor liniare le constituie metoda de eliminare Gauss și metoda de descompunere Cholesky, iar pentru rezolvarea sistemelor neliniare se preferă metoda Newton-Raphson, obținîndu-se valorile funcțiilor sau gradelor de libertate în noduri. Acestea se numesc și necunoscute primare sau de ordinul întîi.

Prezentarea soluției se face de obicei alfa-numeric sub formă de tabele.

f) **efectuarea de calcule suplimentare pentru determinarea necunoscutelor secundare.** În unele probleme, după aflarea necunoscutelor primare, analiza se încheie. Acesta este de obicei cazul problemelor de conducție termică, în care necunoscutele primare sînt temperaturi nodale. În alte probleme, însă, analiza trebuie continuată cu determinarea necunoscutelor secundare sau de ordinul doi. Acestea sînt derivate de ordin superior ale necunoscutelor primare. Și în cazul problemelor termice analiza poate continua cu determinarea necunoscutelor secundare, care sînt intensitățile fluxurilor termice (gradienții termici).

Scanned with OKEN Scanner

3

Modele numerice pentru rezolvarea unor probleme specifice de instalații

3.1. STUDIUL FORMĂRII GHEȚII ȘI AL PREVENIRII ÎNGHEȚĂRII APEI ÎN CONDUCTELE AERIENE SUB PRESIUNE

3.1.1. CONSIDERAȚII PRELIMINARE

În exploatarea pe timp friguros a conductelor aeriene ale instalațiilor de alimentare cu apă și hidroenergetice se pot întîmpla serioase dificultăți datorită atît posibilității de îngheț a apei, astfel încît stratul de gheață format pe peretele interior al conductei să afecteze caracteristicile hidraulice ale curgerii, cît și deformațiilor permanente ale conductelor care apar ca o consecință a acestui fenomen.

Astfel, dacă se întrerupe curgerea apei pentru eventualele intervenții de întreținere și reparații ale sistemului și conducta este plină cu apă, înghețul se propagă într-un anumit interval de timp în toată masa de apă conținută în conductă. Pentru a preveni acest fenomen, în practică există tendința de a se asigura curgerea permanentă prin conductă a unui anumit debit minim. Această precauție, în majoritatea cazurilor, asigură trecerea liberă a debitului prin conductă, însă nu și lipsa totală a gheții pe peretele acesteia, întrucît atunci cînd temperatura apei este aproape 0°C este imposibil de a se evita formarea gheții, chiar pentru viteze depășind 10 m/s.

Se impune deci a se cerceta dacă conducta este capabilă să transporte debitul minim determinat în ipoteza regimului atmosferic nestaționar, fără a afecta caracteristicile hidraulice ale curgerii. Pe de altă parte, chiar și în cazul funcționării normale cu un debit dat, se pune problema cercetării dacă conducta este capabilă să transporte acest debit, ținînd seama de volumul de gheață ce se poate forma în interior, în caz contrar fiind necesar a se prevedea protejarea acesteia prin confecționarea de învelișuri exterioare izolatoare.

În acest context, se expune un model matematic pentru studiul evoluției în timp a grosimii gheții din interiorul conductelor aeriene sub presiune în regim atmosferic nestaționar, care poate servi la găsirea unor soluții economice de protecție împotriva înghețului acestor conducte.

Scanned with OKEN Scanner

3.1.2. DESCRIEREA MODELULUI MATEMATIC

Pentru expunerea teoretică următoare se utilizează notațiile: R — raza interioară a conductei; G — debitul apei în conductă; i_c — panta conductei; H — sarcina hidraulică disponibilă a conductei; ρ_w, c_w, γ_w — densitatea, căldura specifică și greutatea specifică a apei; L — căldura latentă specifică de solidificare a apei; t_w — temperatura apei; J — panta hidraulică; r — raza interioară a gheții; n — coeficientul de rugozitate; ρ_g, λ_g — densitatea și conductivitatea termică a gheții; t_e — temperatura aerului din vecinătatea conductei; w — viteza vîntului; α_i, α_e — coeficienții de convecție interioară și exterioară; τ — timpul.

Din examinarea fenomenelor termice ce au loc în interiorul conductei rezultă că principalele elemente ale energiei termice de care trebuie să se țină seama la efectuarea bilanțului termic, cu referire la un tronson de lungime unitară, sînt:

— căldura transportată de apă Q_w, în J/zi:

$$Q_w = 86400\, G\, \rho_w c_w t_w \tag{3.1}$$

— căldura dezvoltată prin frecare, Q_f, în J/(m zi):

$$Q_f = 86400 \gamma_W G J = \frac{54\,600\pi\gamma_w}{n} r^{2,67} J^{1,5} \tag{3.2}$$

— căldura transmisă de apă, prin peretele conductei, aerului înconjurător Q_p, în J/(m zi):

$$Q_p = \frac{86\,400 \cdot 2\pi(t_1 - t_{NS+1})}{\frac{1}{\alpha_i r_1} + \sum_{j=1}^{NS-1} \frac{1}{\lambda_j} \ln \frac{r_{j+1}}{r_j} + \frac{1}{\alpha_e r_{NS}}} \tag{3.3}$$

Neglijînd rezistența termică a transferului căldurii prin convecție de la apă la conductă (α_i are valori ridicate), precum și rezistența termică a peretelui conductei, datorită influențelor reduse pe care le au asupra fluxului termic și ținînd seama de expresia coeficientului de convecție exterioară [6]:

$$\alpha_e = 3{,}77 \frac{w^{0,7}}{R^{0,3}} \tag{3.4}$$

se determină raza de calcul fictivă R_c ce ține seama de starea de agitație a aerului înconjurător:

$$R_c = R e^{0,615/(wR)^{0,7}}, \tag{3.5}$$

cu care după introducerea temperaturilor $t_1 = 0°C$ și $t_{NS+1} = t_e$, relația (3.3) devine:

$$Q_p = \frac{-86400\pi t_e}{-\frac{1}{2\lambda_g} \ln \frac{r}{R_c}} \tag{3.6}$$

— căldura latentă de solidificare a apei Q_g, în J/m:

$$Q_g = \pi \rho_W (R^2 - r^2) L \tag{3.7}$$

— căldura transmisă de apă gheții Q_t, în J/(m zi) [6]:

$$Q_t = 72 \cdot 10^6 \pi (t_w + 0{,}784 \cdot 10^{-3} H) \left(\frac{G}{r}\right) 0{,}75 \tag{3.8}$$

Scanned with OKEN Scanner

În figura 3.1 s-a reprezentat secțiunea longitudinală printr-o conductă înghețată având lungimea elementară dx, cu elementele termice caracteristice.

Bilanțul termic al ansamblului: apă tranversînd conducta — gheață — mediu exterior, are expresia:

$$-\frac{\partial Q_w}{\partial x}+Q_f-Q_p+\frac{\partial Q_g}{\partial \tau}=0 \tag{3.9}$$

Introducînd notația:

$$r_*=\frac{r}{R_c} \tag{3.10}$$

și ținînd seama de expresia pantei hidraulice J, dedusă din cunoscuta ecuație a debitului Chézy-Manning:

$$J=0{,}255\frac{n^2G^2}{R_c^{5{,}33}}\cdot\frac{1}{r_*^{5{,}33}}, \tag{3.11}$$

de relațiile (3.1), (3.2), (3.6), (3.7) și de ipoteza formării unei mantale cilindrice de gheață pe peretele conductei, adică:

$$\frac{\partial_r}{\partial_x}=0;\quad \frac{\partial_r}{\partial\tau}=\frac{dr}{d\tau} \tag{3.12}$$

ecuația (3.9) devine:

$$-86\,400\,G\rho_w c_w\frac{\partial t_w}{\partial x}+22\,000\frac{\gamma_w n^2G^2}{R_c^{5{,}33}}\cdot\frac{1}{r_*^{5{,}33}}-\frac{86400\pi t_s}{\frac{1}{2\lambda_g}\ln r_*}-2\pi\rho_g LR_c^2r_*\frac{dr}{d\tau}=0 \tag{3.13}$$

Pe de altă parte, numai pentru gheață se poate scrie ecuația de bilanț termic:

$$Q_i-Q_p+\frac{\partial Q_g}{\partial\tau}=0, \tag{3.14}$$

care împreună cu (3.6), (3.7) și (3.8) conduce la relația:

$$72\cdot10^6\pi(t_w+0{,}784\cdot10^{-3}H)\left(\frac{G}{R_c}\right)^{0,75}\frac{1}{r_*^{0,75}}-\frac{86400\pi t_s}{\frac{1}{2\lambda_g}\ln r_*}-2\pi\rho_g LR_c^2r_*\frac{dr_*}{d\tau}=0 \tag{3.15}$$

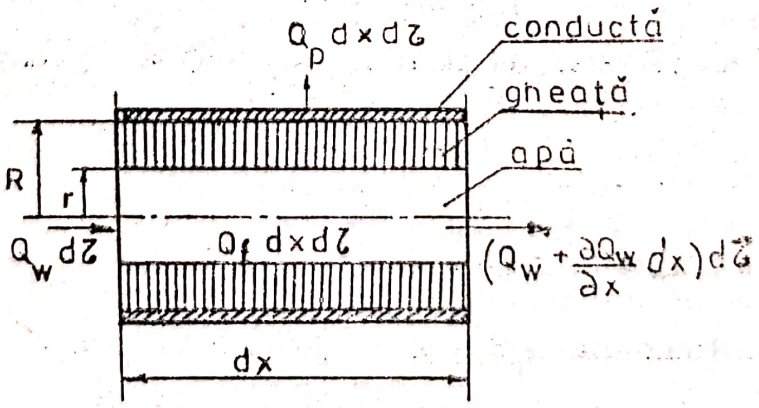


Fig. 3.1. Secțiune longitudinală printr-o conductă înghețată

Scanned with OKEN Scanner

Explicitînd pe t_w din ecuația (3.15), derivînd în raport cu abscisa x și ținînd seama de:

$$\frac{\partial H}{\partial x}=i_c-0{,}255\frac{n^2G^2}{R_c^{5.33}}\frac{1}{r_*^{5.33}}, \tag{3.16}$$

după înlocuirea în (3.13) a valorilor cunoscute rezultă:

$$\frac{dr_*}{d\tau}=0{,}148\frac{Gi_c}{R_c^2}\frac{1}{r_*}+7{,}48\cdot10^{-6}\frac{G^3}{R_c^{7.33}}\frac{1}{r_*^{6.33}}-0{,}66\cdot10^{-3}\frac{t_c}{R_c^2}\frac{1}{r_*\ln r_*}, \tag{3.17}$$

ecuație diferențială care permite studiul evoluției în timp a grosimii gheții în interiorul conductei.

Întrucît temperatura apei este, în general, o mărime necesar a fi cunoscută în practică, aceasta se poate explicita în urma introducerii relației (3.17) în (3.15) și ținînd seama de (3.16):

$$t_w=1{,}256i_cG^{0.25}R_c^{0.75}r_*^{0.75}+0{,}636\cdot10^{-4}\frac{G^{0.25}}{R_c^{4.58}}\frac{1}{r_*^{4.58}}-0{,}784\cdot10^{-6}H \tag{3.18}$$

În ecuația (3.17) derivata $dr_*/d\tau$, pentru anumite condiții date, nu este funcție decît pe raza relativă r_*, încît se pot calcula valorile acestei funcții pentru o serie de valori r_* impuse. Graficul acestei funcții taie axele de coordonate ($dr_*/d\tau=0$) pentru valoarea:

$$r_*=r_{*lim}, \tag{3.19}$$

ce reprezintă raza relativă limită către care tinde gheața din conductă în condițiile date.

Soluția ecuației diferențiale (3.17) se poate obține utilizînd metoda numerică de integrare Runge-Kutta, implementată în cadrul programului ordinator EVGHECA cu schema logică din figura 3.2, unde $\Delta\tau$ este pasul de timp (0,1...0,5 zile), iar ε — eroarea admisă (0,00001...0,001).

Prin formarea gheții pe peretele conductei, coeficientul de rugozitate scade astfel încît, pentru o anumită grosime a gheții caracterizată de raza relativă maxim admisibilă, dată de relația:

$$r_{*ad}=0{,}705\frac{R}{R_e} \tag{3.20}$$

se poate asigura menținerea capacității de transport fără a periclita instalația ca urmare a măririi considerabile a vitezei de curgere a apei în conductă.

Protejarea împotriva înghețului astfel încît să se obțină o conductă lipsită complet de gheață poate conduce la costuri de investiție foarte mari pentru izolație. De aceea, dacă pentru anumite condiții date rezultă că:

$$r_{*lim}\geq r_{*ad}, \tag{3.21}$$

atunci conducta poate să funcționeze fără să fie izolată termic, iar în caz contrar se recomandă să se determine grosimea straturilor ce constituie izolația termică astfel încît în conductă să se formeze gheață cu grosime caracterizată de r_{*ad}, caz în care se va ține seama de relația (3.3).

Punînd $G=0$ în ecuația (3.17), integrînd ecuația astfel obținută cu condiția de margine: $r_*=R/R_i$, pentru $\tau=0$ și considerînd, $r_*=r_{*ad}$, după introducerea relației (3.20), rezultă expresia timpului maxim admisibil de stagnare a apei în conductă τ_{max}, în zile:

$$\tau_{max}=758\frac{R^2}{t_c}\left(0{,}503\ln\frac{R}{R_e}-0{,}076\right) \tag{3.22}$$

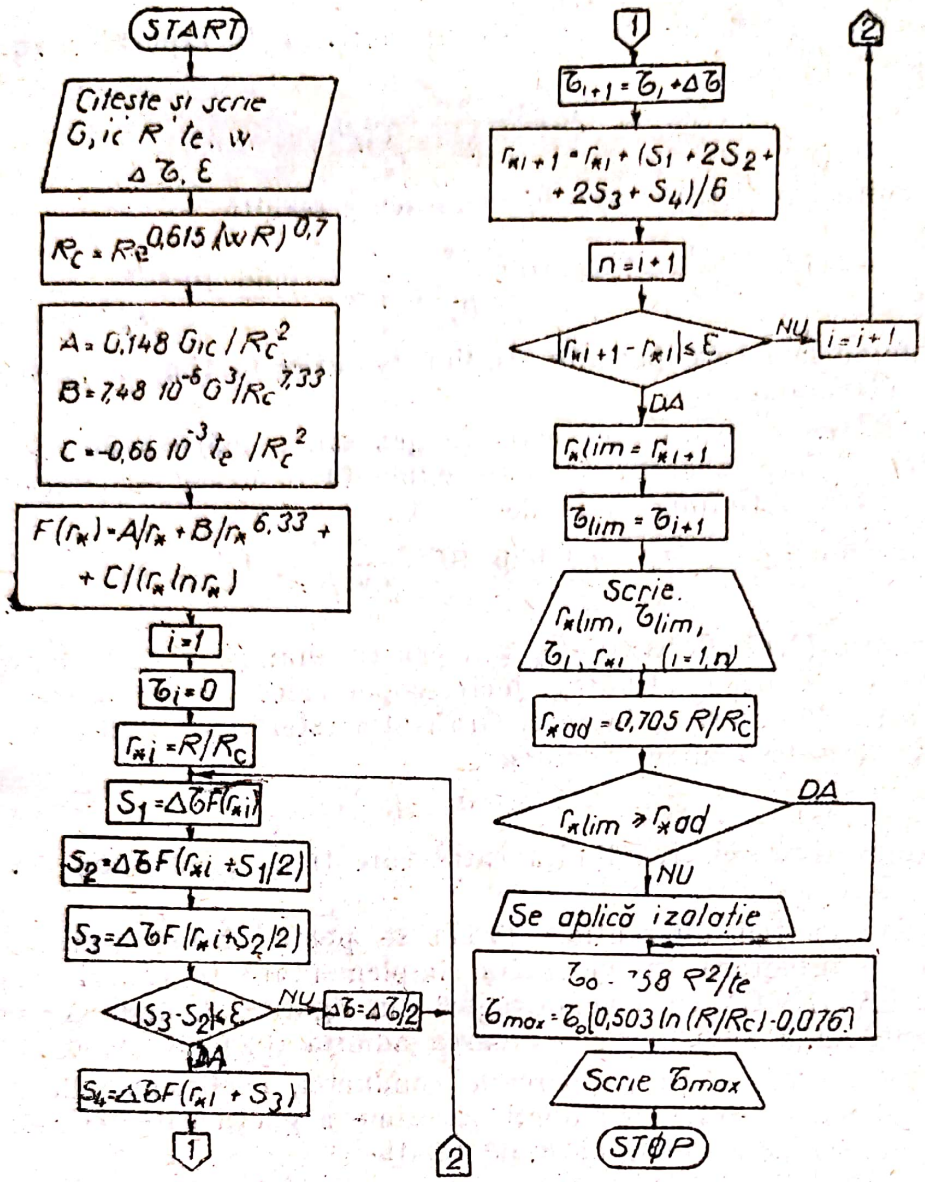


Fig. 3.2. Schema logică a modelului de calcul al formării gheții și protecției conductelor aeriene sub presiune

3.2. ANALIZA REȚELELOR INELARE DE DISTRIBUȚIE A APEI

3.2.1. CONSIDERAȚII PRELIMINARE

Alimentarea cu apă a centrelor populate și industriilor se efectuează prin rețele de distribuție, care se realizează la dimensiuni din ce în ce mai mari și cu o structură tot mai complexă (rețele inelare, alimentare din mai multe surse, stații de pompare integrate, elemente potențiale). Pe de altă parte, proiectarea extinderilor sau reconsiderarea unor rețele în vederea optimizării funcționării acestora necesită determinarea repartiției debitelor și a distribuției presiunilor în aceste rețele.

Aceste considerente impun formularea unor modele matematice adecvate, care să permită determinarea regimului hidraulic în astfel de rețele inelare, avînd o deosebită importanță în practica analizării lor.

Scanned with OKEN Scanner

Ultimele variante ale metodologiei de calcul a rețelelor de distribuție a apei în regim permanent au evoluat mult, apelînd la noțiuni de teoria grafurilor, calcul matriceal, programare liniară și neliniară, în corelație cu posibilitățile noi oferite de calculatoarele electronice.

În funcție de mărimile hidraulice adoptate ca necunoscute, s-au conturat trei categorii de modele matematice: cele care consideră cotele piezometrice în noduri (*metoda nodurilor*), cele care consideră debitele de corecție pe inele (*metoda ciclurilor*) și respectiv cele care consideră debitele pe tronsoane (*metoda tronsoanelor*) drept necunoscute. Întrucît sistemul de ecuații astfel formulat este neliniar, procedeele de rezolvare sînt în general iterative.

Deși în formularea nodală numărul de ecuații este mai mare decît în formularea ciclică, care oferă o bună convergență și stabilitate numerică a procesului iterativ, densitatea elementelor nenule ale matricelor ecuațiilor nodale este mai mică decît cea pentru ecuațiile pe inele. Totodată, ecuațiile nodale sînt mai ușor de formulat, ceea ce face ca metoda nodurilor să se preteze bine la analiza rețelelor de distribuție.

În acest context, se descrie un model matematic de analiză cu grad sporit de generalitate, ce permite calculul regimului hidraulic în rețele inelare de distribuție a apei în regim permanent bazat pe metoda nodurilor, în cadrul căreia se utilizează pentru rezolvarea sistemului neliniar de ecuații nodale, algoritmul iterativ Newton-Raphson.

3.2.2. BAZELE CALCULULUI HIDRAULIC AL UNEI REȚELE INELARE ÎN REGIM PERMANENT DE CURGERE A APEI

Oricărei rețele de distribuție îi corespunde un graf orientat asociat, compus din arce care pot reprezenta tronsoane de conducte, pompe sau armături și din vîrfuri, ca rezervoare sau intersecții de conducte.

Pentru o rețea cu un singur nod cu cotă piezometrică cunoscută, avînd T tronsoane, N noduri și M inele, pe baza ecuației lui Euler, care dă numărul ciclomatic al unui graf, se poate scrie:

$$T=M+N-1 \tag{3.23}$$

O rețea inelară este echilibrată din punct de vedere hidraulic cînd sînt satisfăcute condițiile:

— continuitatea debitului în noduri:

$$f_j=\sum_{\substack{i\neq j\\ i=1}}^{N} Q_{ij}+q_j=0 \qquad (j=1,\ldots,N-1) \tag{3.24}$$

în care f_j este debitul rezidual din nodul j; Q_{ij} — debitul tronsonului ij cu semnul (+) cînd intră în nodul j, respectiv cu semnul (—) cînd iese din nodul j; q_j — debitul concentrat al nodului j, cu semnul (+) cînd alimentează nodul și cu semnul (—) cînd se consumă în nod.

În particular, pentru rețeaua ramificată ($M=0$; $T=N-1$) numărul ecuațiilor de nod egal cu $N-1$ este suficient pentru determinarea debitelor necunoscute.

— conservarea energiei pe inele:

$$f_m=\sum_{\substack{ij\in m\\ ij=1}}^{T} \varepsilon_{ij} h_{ij}=0 \qquad (m=1,\ldots,M) \tag{3.25}$$

Scanned with OKEN Scanner

unde: f_m este pierderea de sarcină reziduală (divergenţa) în inelul m; e_{ij} — orientarea tronsonului ij cu valoarea (+1) sau (—1) după cum sensul de curgere a apei în tronsonul ij este acelaşi sau opus cu sensul de parcurs al inelului m; h_{ij} — pierderea de sarcină longitudinală pe tronsonul ij, ce este legată de debit prin relaţia funcţională generală:

$$h_{ij} = S_{ij} Q_{ij}^{\beta} \tag{3.26}$$

în care: S_{ij} este modulul de rezistenţă hidraulică al tronsonului ij; β — exponent cu valoarea 1,8...2.

Pentru calculul hidraulic se dispune de un număr T de ecuaţii dat de relaţia (3.23), care pot avea ca necunoscute debitele sau pierderile de sarcină pe tronsoane. În ambele cazuri intră implicit ca necunoscute şi diametrele tronsoanelor, prin intermediul rezistenţei hidraulice S_{ij}, fiind necesar ca acestea să se determine printr-un calcul de predimensionare dacă nu sînt cunoscute ca la analiza reţelei.

3.2.3. DESCRIEREA MODELULUI MATEMATIC

Modelul de calcul presupune cunoscute următoarele date de bază: topologia reţelei; lungimile, diametrele şi rugozitatea absolută ale conductelor reţelei; cotele geodezice şi debitele concentrate pentru fiecare nod; cota piezometrică într-unul din nodurile reţelei (punct obligat sau instalaţie de punere sub presiune).

3.2.3.1. Principiul metodei nodurilor

În cazul aplicării metodei nodurilor pentru echilibrarea reţelei, se folosesc relaţiile (3.24) şi (3.26).

Relaţia funcţională (3.26) se pune sub forma:

$$Q_{ij} = S_{ij}^{-1/\beta} h_{ij}^{1/\beta} = S_{ij}^{-1/\beta} (Z_i - Z_j) \, |Z_i - Z_j|^{(1-\beta)/\beta} \tag{3.27}$$

în care Z_i şi Z_j sînt cotele piezometrice în nodurile i şi j.

Introducînd (3.27) în (3.24), ecuaţia de continuitate a debitului devine:

$$f_j = \sum_{\substack{i=1 \\ i \neq j}}^{N} S_{ij}^{-1/\beta} (Z_i - Z_j) \, |Z_i - Z_j|^{(1-\beta)/\beta} + q_j = 0 \quad (j = 1, \ldots, N-1) \tag{3.28}$$

Relaţia (3.28) scrisă pentru toate cele N noduri conduce la un sistem de $N-1$ ecuaţii neliniare independente, prin rezolvarea căruia se determină cotele piezometrice în $N-1$ noduri, în unul din noduri cota piezometrică fiind necesară a fi cunoscută.

Principiul metodei nodale constă în considerarea unui set de cote piezometrice iniţiale în noduri ca şi cunoscute, care se corectează succesiv pînă cînd sînt satisfăcute condiţiile (3.28).

Pentru a realiza acest obiectiv, se pretează bine utilizarea algoritmului de rezolvare numerică Newton-Raphson, care efectuează corecţia simultană la toate nodurile.

Scanned with OKEN Scanner

3.2.3.2. Formularea și aplicarea algoritmului Newton-Raphson

Se consideră o aproximare inițială a cotelor piezometrice $Z_0=\{Z_1^{(0)}\ Z_2^{(0)}\ldots Z_{N-1}^{(0)}\}$ și vectorul de corecție $\Delta Z_{k+1}=\{\delta Z_1^{(k+1)}\ldots\delta Z_{N-1}^{(k+1)}\}$, la fiecare pas $k+1$ al procesului iterativ:

$$Z_{k+1}=Z_k+\Delta Z_{k+1}, \tag{3.29}$$

ce se obține rezolvînd sistemul:

$$J_k\Delta Z_{k+1}=-F(Z_k) \tag{3.30}$$

unde J_k este matricea Jacobian corespunzătoare vectorului Z_k.

Dezvoltat sistemul (3.30) primește forma:

$$\begin{vmatrix} \frac{\partial f_1}{\partial Z_1} & \cdots & \frac{\partial f_1}{\partial Z_{N-1}} \\ \vdots & & \vdots \\ \frac{\partial f_{N-1}}{\partial Z_1} & \cdots & \frac{\partial f_{N-1}}{\partial Z_{N-1}} \end{vmatrix} \begin{Bmatrix} \delta Z_1 \\ \vdots \\ \partial Z_{N-1} \end{Bmatrix} = \begin{Bmatrix} -f_1 \\ \vdots \\ -f_{N-1} \end{Bmatrix} \tag{3.31}$$

în care derivatele parțiale se obțin din (3.28) după cum urmează:

$$\frac{\partial f_j}{\partial Z_i}=\frac{\partial f_i}{\partial Z_j}=\frac{1}{\beta}S_{ij}^{-1/\beta}(Z_i-Z_j)^{(1-\beta)/\beta};\quad i\neq j \tag{3.32}$$

$$\frac{\partial f_j}{\partial Z_j}=-\sum_{\substack{i\neq j\\ i=1}}^{N}\frac{\partial f_j}{\partial Z_i} \tag{3.33}$$

Din examinarea relației (3.28) se observă că satisfacerea continuității debitelor în noduri se poate realiza admițînd ca variabile atît cotele piezometrice în noduri Z_j și rezistențele hidraulice ale tronsoanelor S_{ij}, cît și debitele concentrate în noduri q_j, cu condiția ca suma tuturor acestor necunoscute să fie $N-1$, iar utilizarea modelului de calcul poate fi extinsă la rezolvarea de noi probleme. Astfel, dacă se notează necunoscutele cote piezometrice cu $\bar{Z}=\{Z_1\ldots Z_w\}$, rezistențe hidraulice cu $\bar{S}=\{S_{ij}\ldots S_{pr}\}$ și debite concentrate în noduri cu $\bar{q}=\{q_j\ldots q_t\}$, atunci pornind de la un vector inițial al necunoscutelor $Z_0=\{Z_1^{(0)}\ldots Z_w^{(0)}\,S_{ij}^{(0)}\ldots S_{pr}^{(0)}\,q_j^{(0)}\ldots q_t^{(0)}\}$ se determină corecțiile la fiecare iterație din sistemul:

$$\begin{vmatrix} \frac{\partial f_1}{\partial Z_1} & \cdots & \frac{\partial f_1}{\partial q_t} \\ \vdots & & \vdots \\ \frac{\partial f_{w+1}}{\partial Z_1} & \cdots & \frac{\partial f_{w+1}}{\partial q_t} \\ \vdots & & \vdots \\ \frac{\partial f_{N-1}}{\partial Z_1} & \cdots & \frac{\partial f_{N-1}}{\partial q_t} \end{vmatrix} \begin{Bmatrix} \delta Z_1 \\ \vdots \\ \delta S_{ij} \\ \vdots \\ \delta q_t \end{Bmatrix} = \begin{Bmatrix} -f_1 \\ \vdots \\ -f_{w+1} \\ \vdots \\ -f_{N-1} \end{Bmatrix} \tag{3.34}$$

În care derivatele parțiale referitoare la S_{ij} și q_j au expresiile:

$$\frac{\partial f_j}{\partial S_{ij}}=-\frac{1}{\beta}S_{ij}^{-(1+\beta)/\beta}(Z_i-Z_j)^{1/\beta} \tag{3.35}$$

$$\frac{\partial f_j}{\partial q_j}=1 \tag{3.36}$$

Scanned with OKEN Scanner

Procesul iterativ se încheie la atingerea preciziei de calcul stabilite prin condiția:

$$\left|-f_j^{(k)}\right| \leqslant \varepsilon \qquad (j=1,\ldots,N-1) \tag{3.37}$$

unde ε este eroarea admisă.

Cunoscînd cotele piezometrice în noduri, se determină debitele de tranzit în conducte cu ajutorul relației (3.27).

Sistemul de ecuații (3.34) nu are soluție pentru orice combinație a necunoscutelor, la alegerea acestora trebuind să se respecte unele reguli, rezultate din neadmiterea existenței de linii sau coloane în matricea sistemului cu toți termenii nenuli, după cum urmează:

— la fiecare nod trebuie să existe cel puțin una din necunoscute: debit concentrat în nod, cotă piezometrică în nodul respectiv sau la orice nod adiacent, rezistență hidraulică a oricărui tronson care concură în nod;

— un nod care are debitul concentrat necunoscut trebuie conectat la cel puțin un alt nod cu debit cunoscut;

— un tronson de conductă cu rezistența hidraulică necunoscută nu trebuie să aibă mai mult de o necunoscută la nodurile care îl definesc, ce poate fi cota piezometrică sau debitul.

Datorită structurii sistemului neliniar de ecuații (3.28), pentru aplicarea algoritmului Newton-Raphson se ține seama de următoarele avantaje:

— matricea Jacobian, cu care se efectuează operații repetate de calcul conține cel mult $N-1+2T$ elemente nenule ($N-1$ elemente diagonale și $2T$ elemente nediagonale) ceea ce îi conferă proprietatea de a fi rară;

— în majoritatea cazurilor, această matrice este simetrică ireductibilă și slab dominant diagonală, ceea ce asigură existența matricei inverse;

— mai mult, matricea inversă este o matrice pozitivă, proprietate ce conferă calități de stabilitate numerică în rezolvarea sistemului algebric liniar (3.34) în fiecare iterație a algoritmului Newton-Raphson.

Pe baza acestui model de analiză nodală, generalizat pentru rețele de distribuție alimentate din mai multe surse și cu pompe integrate pe artere, s-a elaborat programul ordinator ANOREC, în cadrul căruia s-au luat unele măsuri de evitare a punctelor singulare ($Z_i=Z_j$) în Jacobianul sistemului de ecuații nodale.

Avînd în vedere posibilitatea de a introduce ca necunoscute pe lîngă cote piezometrice și rezistențele hidraulice S_{ij} și debitele concentrate q_j, metoda nodală oferă elasticitate mai mare, pretîndu-se în special pentru analiza rețelelor complexe de distribuție a apei, care în general conține elemente ce prezintă o relație funcțională pierdere de sarcină —debit continuă.

3.3. CALCULUL REGIMULUI HIDRAULIC ÎN REȚELE INELARE COMPLEXE DE DISTRIBUȚIE A ENERGIEI TERMICE

3.3.1. CONSIDERAȚII PRELIMINARE

Rețelele de termoficare se realizează la dimensiuni din ce în ce mai mari (cu tot mai multe conducte, puncte termice etc.) și cu o structură mereu mai complexă (rețele inelare, alimentare cu agent termic din mai multe surse, stații intermediare de pompare). Existența conductelor de tur și retur, care pot avea caracteristici geometrice diferite sau pot fi echipate în mod diferit cu stații intermediare de pompare, complică și mai mult o rețea de termoficare.

Scanned with OKEN Scanner

Formularea unor modele matematice adecvate, care să permită determinarea circulației debitelor și distribuției presiunilor în astfel de rețele este esențială atît pentru rezolvarea corectă și eficientă a etapei de proiectare, cît și pentru analiza rețelei în diferite regimuri de exploatare, normale sau de avarie.

Astfel, cunoscînd topologia rețelei, caracteristicile conductelor, consumul fiecărui utilizator, producția fiecărei centrale, precum și presiunea necesară într-un punct obligat al rețelei, se pune problema determinării debitelor de curgere pe tronsoane și a presiunilor în nodurile de consum.

Această problemă apare în particular la proiectarea extinderilor sau reconsiderarea unor rețele în vederea optimizării funcționării lor.

În acest context, se descrie un model matematic pentru calculul regimului hidraulic în rețele inelare complexe de distribuție a energiei termice, bazat pe metoda ciclurilor, în cadrul căreia se utilizează pentru rezolvarea sistemului neliniar de ecuații pe inele un algoritm iterativ.

3.3.2. BAZELE CACULULUI HIDRAULIC AL UNEI REȚELE TERMICE

Oricărei rețele termice îi corespunde un graf neorientat asociat, compus din muchii care pot reprezenta tronsoane de conducte sau pompe, și din vîrfuri, ca centrale de termoficare, utilizatori sau intersecții de conducte.

O rețea este inelară dacă graful neorientat asociat ei conține cel puțin un ciclu. Un inel este deci format dintr-un ciclu de conducte de tur, ce nu conține conexiuni între conductele de tur și cele de retur. Acestui inel îi corespunde un altul format cu conductele corespunzătoare de retur.

O rețea este echilibrată din punct de vedere hidraulic cînd sînt satisfăcute condițiile:

— continuitatea debitului în noduri:

$$\sum_{\substack{i=1\\ i\neq j}}^{N} G_{ij} + q_j = 0 \qquad (j=1,\ldots,N-1) \tag{3.38}$$

n care: G_{ij} este debitul tronsonului ij cu semnul $(+)$ cînd intră în nodul j respectiv cu semnul $(-)$ cînd iese din nodul j; q_j — debitul concentrat al nodului j, cu semnul $(+)$ cînd alimentează nodul și cu semnul $(-)$ cînd se consumă în nod.

— conservarea energiei pe inele:

$$\sum_{\substack{ij=1\\ ij\in m}}^{T} \varepsilon_{ij}\Delta p_{ij} = 0 \qquad (m=1,\ldots,M) \tag{3.39}$$

unde ε_{ij} este orientarea tronsonului ij cu valoarea $(+1)$ sau (-1) după cum sensul de curgere a apei în conducta ij corespunde sau nu cu sensul de parcurs al inelului m; Δp_{ij} — diferența de presiune între cele două extremități ale tronsonului ij, ce este legată de debit prin relația funcțională generală:

$$\Delta p_{ij} = p_i - p_j = S_{ij}|G_{ij}|G_{ij} + \gamma(ZT_j - ZT_i) - \Pi_{ij} \tag{3.40}$$

Scanned with OKEN Scanner

în care: ZT_i, ZT_j sînt cotele geodezice în nodurile i și j; γ — greutatea specifică a apei; Π_{ij} — presiunea activă a stației de pompare intermediare de pe tronsonul ij (în cazul că aceasta există) independentă de debit; S_{ij} — rezistența hidraulică a tronsonului ij, dată de relația:

$$S_{ij}=\frac{8\rho}{\pi^2}\left(\frac{\lambda_{ij}L_{ij}}{D_{ij}}+\zeta_{ij}\right)\frac{1}{D_{ij}^4}, \tag{3.41}$$

cu notațiile: ρ — densitatea apei; L_{ij}, D_{ij} — lungimea și diametrul tronsonului ij; λ_{ij}, ζ_{ij} — coeficientul de rezistență hidraulică longitudinală și respectiv locală a tronsonului ij.

Presupunînd că ecuațiile (3.38) și (3.39) se referă la mișcarea apei în conductele de tur, relații analoage se pot scrie și pentru curgerea apei în conductele rețelei de retur:

$$\sum_{\substack{i\neq j\\ ij=1}}^{N}\bar{G}_{ij}+\bar{q}_j=0 \qquad (j=1,\ldots,N-1) \tag{3.42}$$

$$\sum_{\substack{ij\in m\\ ij=1}}^{T}\varepsilon_{ij}\Delta\bar{p}_{ij}=0 \qquad (m=1,\ldots,M) \tag{3.43}$$

unde:

$$\Delta\bar{p}_{ij}=\bar{p}_i-\bar{p}_j=\bar{S}_{ij}|\bar{G}_{ij}|\bar{G}_{ij}+\gamma(ZT_j-ZT_i)-\bar{\Pi}_{ij} \tag{3.44}$$

Notînd:

$$c_{ij}=\gamma(ZT_j-ZT_i)-\Pi_{ij};\ \bar{c}_{ij}=\gamma(ZT_j-ZT_i)-\bar{\Pi}_{ij}, \tag{3.45}$$

relațiile funcționale (5.40) și (5.44) devin:

$$p_i-p_j=S_{ij}|G_{ij}|G_{ij}+c_{ij} \tag{3.46}$$

$$\bar{p}_i-\bar{p}_j=\bar{S}_{ij}|\bar{G}_{ij}|\bar{G}_{ij}+\bar{c}_{ij} \tag{3.47}$$

Diferența de presiune între conducta de tur și de retur într-un nod j trebuie să satisfacă relația:

$$p_j-\bar{p}_j\geqslant\delta_j \qquad (j=1,\ldots,N) \tag{3.48}$$

în care δ_j este diferența minimă de presiune ce asigură curgerea debitului q_j prin instalația consumatorului racordat la nodul j.

Pentru calculul de echilibrare hidraulică a unei rețele termice inelare de tur sau de retur, cu un singur nod cu cota piezometrică cunoscută, se dispune de un număr T de ecuații dat de relația:

$$T=M+N-1 \tag{3.49}$$

în care: T este numărul tronsoanelor; M — numărul inelelor independente; N — numărul nodurilor.

3.3.3. DESCRIEREA MODELULUI MATEMATIC

Modelul de calcul presupune cunoscute următoarele date de bază: topologia rețelei; caracteristicile geometrice și hidraulice ale conductelor; cotele geodezice și debitele concentrate pentru fiecare nod; presiunile active introduse prin stații intermediare de pompare; cota piezometrică într-unul din nodurile rețelei.

Scanned with OKEN Scanner

Sistemul de ecuații neliniare (3.38), (3.46) se compune din $N-1$ ecuații de tipul (3.38) și T ecuații de tipul (3.46) pentru determinarea celor T debite în conducte și $N-1$ presiuni în nodurile rețelei. Acesta se reduce la un sistem cu dimensiunea egală cu numărul inelelor independente M dat de relația (3.49), care se rezolvă prin aproximații succesive.

Pentru fiecare inel independent m se alege un sens de parcurs și i se asociază un debit de corecție G_m.

Pornind de la o repartiție inițială a debitelor $G_{ij}^{(0)}$, care satisface relația (3.38), acestea se corectează folosind relația:

$$G_{ij}=G_{ij}^{(0)}+\sum_{\substack{ij\in m\\ m=1}}^{M}\varepsilon_{ij}G_m \qquad (ij=1,\ldots,T) \tag{3.50}$$

în care ε_{ij} are semnificația din relația (3.39), iar debitele de corecție $G_m(m=1,\ldots,M)$ se determină rezolvînd iterativ sistemul de ecuații (3.39).

Cunoscînd soluția aproximativă $G_m^{(k)}(m=1,\ldots,M)$ obținută la iterația k, se caută printre ecuațiile (3.39) pe cea cu divergența cea mai mare. Fie $f_{m_*}(G_{ij})=0$ această ecuație și $f_{m_*}(G_{ij}^{(k)})$ valoarea sa. Se definește corecția de debit:

$$G_m^{(k+1)}=\begin{cases}G_m^{(k)}, \text{ pentru } m\neq m_*;\\ G_m^{(k)}+\delta G_m, \text{ pentru } m=m_*\end{cases} \tag{3.51}$$

cu δG_{m_*} exprimat prin relația:

$$f_{m_*}(G_{ij}^{(k)})+\delta G_{m_*}\cdot\left.\frac{\partial f_{m_*}}{\partial G_{m_*}}\right|_{G_{ij}^{(k)}}=0, \tag{3.52}$$

de unde rezultă formula de corecție cunoscută în aproximația liniară:

$$\delta G_{m_*}=-\frac{\sum\limits_{(m_*)}\varepsilon_{ij}\left(S_{ij}\left|G_{ij}^{(k)}\right|G_{ji}^{(k)}+c_{ij}\right)}{2\sum\limits_{(m_*)}S_{ij}\left|G_{ij}^{(k)}\right|} \tag{3.53}$$

Procesul iterativ se oprește atunci cînd fiecare din ecuațiile (3.39) sînt satisfăcute cu o precizie impusă.

Această metodă este eficace întrucît modificarea debitelor în conductele unui inel modifică în mod egal echilibrul celorlalte inele avînd conducte comune cu acesta.

Pentru a accelera convergența modelului de calcul se determină rețeaua ramificată optimă de bază, folosind algoritmul de optimizare descris la punctul 5.2.1., care minimizează suma rezistențelor hidraulice S_{ij}.

Soluția inițială $G_{ij}^{(0)}$ se obține considerînd debitele nule în conductele care nu aparțin rețelei ramificate de bază și determinînd celelalte debite recursiv, pornind de la nodurile extreme ale acestei rețele.

Avînd astfel obținută soluția sistemului (3.38), (3.46), se aplică același model de calcul iterativ pentru determinarea soluției sistemului (3.42), (3.47). În general, o bună soluție inițială $\bar{G}_{ij}^{(0)}$ pentru acest sistem este $\bar{G}_{ij}^{(0)}=-G_{ij}$ $(ij=1,\ldots,T)$.

După ce s-a determinat repartiția finală a debitelor, se calculează cu relațiile funcționale (3.40), (3.44) pierderile de presiune pe tronsoane, apoi pornind de la un nod cu presiunea cunoscută se obțin presiunile în toate celelalte noduri.

Scanned with OKEN Scanner

3.4. ANALIZA NUMERICĂ A PROCESULUI POMPĂRII APEI ÎN SISTEMELE DE ALIMENTARE CU APĂ ŞI CĂLDURĂ

3.4.1. CONSIDERAŢII PRELIMINARE

Distribuţia apei la utilizatori într-un sistem de alimentare cu apă sau cu căldură se realizează printr-o reţea de conducte, de obicei cu ajutorul unor staţii de pompare.

Luînd în considerare faptul că staţiile de pompare se caracterizează printr-un regim de exploatare în care debitul variază, se impune ca pompele să fie capabile de a-şi regla debitele în cursul funcţionării zilnice. Cea mai frecvent utilizată schemă este funcţionarea în paralel a mai multor pompe cu turaţie constantă, dar există şi procedee moderne care folosesc pompe cu turaţie variabilă şi diverse combinaţii.

Pentru analiza pompării apei în sistemele de alimentare cu apă şi căldură, care urmăreşte determinarea punctelor de funcţionare ale pompelor exploatate în diferite regimuri, se prezintă un model de calcul numeric ce poate fi utilizat atît pentru pompele cu turaţie constantă cît şi pentru cele cu turaţie variabilă.

3.4.2. DESCRIEREA MODELULUI MATEMATIC

Cunoscînd caracteristicile pompelor, topologia şi caracteristica reţelei precum şi variaţia consumurilor de apă se pot stabili punctele de funcţionare a sistemului staţie de pompare—reţea de distribuţie folosind un model de calcul numeric pentru analiza funcţionării staţiilor de pompare.

Pornind de la un set iniţial de date care reprezintă curba caracteristică a pompelor (înălţime de pompare H — debit pompat Q) prin puncte $y_k = f(x_k)$, se generează curba continuă H-Q utilizînd un algoritm de interpolare cubică care solicită determinarea unei funcţii aproximative $F(x)$ astfel încît în orice nod de interpolare x_k să fie satisfăcută condiţia $f(x_k)=F(x_k)$.

Dacă interpolarea se face pe subintervale $[x_i, x_{i+1}]$ ale intervalului globa de interpolare $[x_0, x_n]$ avem de a face cu funcţii spline. Funcţia aproximativă $F_i(x)$ se consideră sub forma unui polinom algebric de gradul III, avînd expresia:

$$F_i(x)=c_0+c_1x+c_2x^2+c_3x^3 \tag{3.54}$$

în care $c_0 \ldots c_3$ sînt coeficienţi necunoscuţi, iar $x_i<x<x_{i+1}$.

Pe un interval întreg se utilizează ca funcţie de aproximare un set de funcţii cubice $F_i(x)$.

Pentru a determina expresia funcţiei cubice de aproximare $g(x)=F_i(x)$ pentru intervalul i se va rezolva sistemul:

$$\begin{gathered} f(x_i)=c_0+c_1x_i+c_2x_i^2+c_3x_i^3 \\ f(x_{i+1})=c_0+c_1x_{i+1}+c_2x_{i+1}^2+c_3x_{i+1}^3 \\ g''(x_i)=f(x_i) \\ g''(x_{i+1})=f(x_{i+1}) \end{gathered} \tag{3.55}$$

Scanned with OKEN Scanner

Expresia matematică obținută în urma efectuării calculelor este de forma:

$$F_i(x)=f(x_{i+1})\frac{x-x_i}{x_i}+f(x_i)\frac{x_{i+1}-x}{x_i}+\frac{g''(x_{i+1})}{6}\left[\frac{(x-x_i)^3}{\Delta x_i}-\Delta x_i(x-x_i)\right]+$$

$$+\frac{g''(x_i)}{6}\left[\frac{(x_{i+1}-x)^3}{\Delta x_i}-\Delta x_i(x_{i+1}-x)\right] \qquad (3.56)$$

Considerînd că se folosesc puncte echidistante $\Delta x_i=\Delta x$, pentru calcularea lui $g''(x)$ care apare în relația (3.56) se aplică condițiile la limită:

— de aproximare după pantă în x_i și x_{i+1}:

$$F_i'(x_i)=F_{i-1}'(x_i) \qquad (3.57)$$

— de aproximare după curbură în x_i:

$$F_i''(x_i)=F_{i-1}''(x_i) \qquad (3.58)$$

În urma efectuării derivărilor impuse de condițiile la limită, acestea conduc la un sistem de ecuații liniare de forma (2.20) a cărui rezolvare numerică se face prin metoda Gauss, rezultînd valorile $g''(x_i)$ pentru toate intervalele încît pentru orice $x=Q$ dat să se poată calcula valoarea corespunzătoare a lui $F_i(x)=H$, folosind expresia (3.56).

Pe baza caracteristicilor geometrice și hidraulice ale sistemului de conducte (materialul, lungimea și coeficienții de rezistență hidraulică ai conductelor, nivelul static, intervalele de variație ale debitelor) se determină curba sarcină de pompare — caracteristică hidrodinamică a rețelei (H-S).

Curba caracteristică a rețelei de conducte exprimă înălțimea de pompare funcție de debit ca sumă a înălțimii statice și a pierderilor de sarcină liniare și locale (rezistența hidrodinamică a rețelei):

$$H=H_s+SQ^2 \qquad (3.59)$$

în care: H este sarcina de pompare; H_s — înălțimea statică; S — caracteristica hidrodinamică totală a rețelei; Q — debitul pompat.

Deoarece nivelul apei în camera de aspirație variază între limitele corespunzătoare pornirii și opririi pompei, înălțimea statică variază și ea determinînd o familie de curbe paralele H-S. Curba superioară corespunde nivelului minim în camera de aspirație, iar curba inferioară corespunde nivelului maxim în această cameră.

Punctul de funcționare al pompei se găsește la intersecția curbelor H-Q și S-Q, determinîndu-se folosind o metodă iterativă în cadrul căreia ecuația algebrică (3.59) se rezolvă prin metoda Newton. Primul pas în determinarea acestuia constă în aflarea prin aproximații succesive a unui mic interval în interiorul căruia se intersectează cele două curbe, urmat de aproximarea liniară a curbei H-Q pe acest interval cu o expresie de forma:

$$H=a_1Q+a_2 \qquad (3.60)$$

pentru $H_{k+1}\leqslant H\leqslant H_k$.

Aplicînd relația (3.60) pentru limitele intervalului $[k,\ k+1]$:

$$\begin{aligned} H_k&=a_1Q_k+a_2 \\ H_{k+1}&=a_1Q_{k+1}+a_2 \end{aligned} \qquad (3.61)$$

Scanned with OKEN Scanner

rezultă prin rezolvarea sistemului de ecuații (3.61) coeficienții:

$$a_1=\frac{H_k-H_{k+1}}{Q_k-Q_{k+1}};\qquad a_2=\frac{H_{k+1}Q_k-H_kQ_{k+1}}{Q_k-Q_{k+1}} \tag{3.62}$$

cu care, în final, expresia (3.60) primește forma:

$$H=\frac{H_k-H_{k+1}}{Q_k-Q_{k+1}}Q+\frac{H_{k+1}Q_k-H_kQ_{k+1}}{Q_k-Q_{k+1}}, \tag{3.63}$$

iar ecuația ce se obține, ținînd seama de (3.59) se poate rezolva cu metoda Newton.

Modelul de calcul numeric descris permite analiza asistată de calculator a procesului de pompare, fiind de real folos în proiectarea și exploatarea sistemelor de alimentare cu apă și căldură.

3.5. MODELAREA MATEMATICĂ A PROCESELOR DE TRANSFER TERMIC DIN REZERVOARELE DE ACUMULARE ALE INSTALAȚIILOR DE PREPARARE A APEI CALDE

3.5.1. IPOTEZE DE CALCUL

Acumularea energiei termice în medii lichide reprezintă cea mai utilizată modalitate de stocare a căldurii, atît în cadrul instalațiilor clasice de preparare a apei calde cît și a sistemelor cu surse neconvenționale de energie.

Modelarea proceselor de transfer termic ce au loc în lichidul din rezervorul de acumulare se face admițînd două ipoteze simplificatoare:

— pierderile de căldură ale rezervorului de stocaj se consideră neglijabile;

— circulația apei prin rezervor este uniformă.

Dacă rezervoarele de acumulare diurne avînd capacități de 5...50 m³ pot fi considerate fără pierderi de căldură la numai o izolație de 5 cm vată minerală, în schimb în cazul rezervoarelor de lungă durată pentru a putea admite aceeași ipoteză este necesară o capacitate de peste 50000 m³ cu o izolație termică de 25 cm vată minerală.

Uniformitatea circulației poate fi aproximativ asigurată practic, printr-o introducere cît mai uniformă a apei în rezervor asociată cu existența la partea superioară a unor dispozitive cu rol de uniformizare.

3.5.2. MODELE MATEMATICE

a) Ipotezele de calcul admise permit, pentru stabilirea modelului matematic în *metoda „caracteristicilor"*, considerarea unui element infinit mic desfășurat pe toată secțiunea transversală a rezervorului (fig. 3.3.).

Bilanțul termic al elementului infinit mic este:

$$QSd\tau+Gctd\tau=\left(Q+\frac{\partial Q}{\partial x}dx\right)Sd\tau+Gc\left(t+\frac{\partial t}{\partial x}dx\right)dx+Sds\rho c\frac{\partial t}{\partial \tau}d\tau \tag{3.64}$$

în care:

$$G=Sw\rho \tag{3.65}$$

$$Q=-\lambda\frac{\partial t}{\partial x}\Big|_x \tag{3.66}$$

$$Q+\frac{\partial Q}{\partial x}dx=-\lambda\frac{\partial t}{\partial x}\Big|_{x+dx} \tag{3.67}$$

Scanned with OKEN Scanner

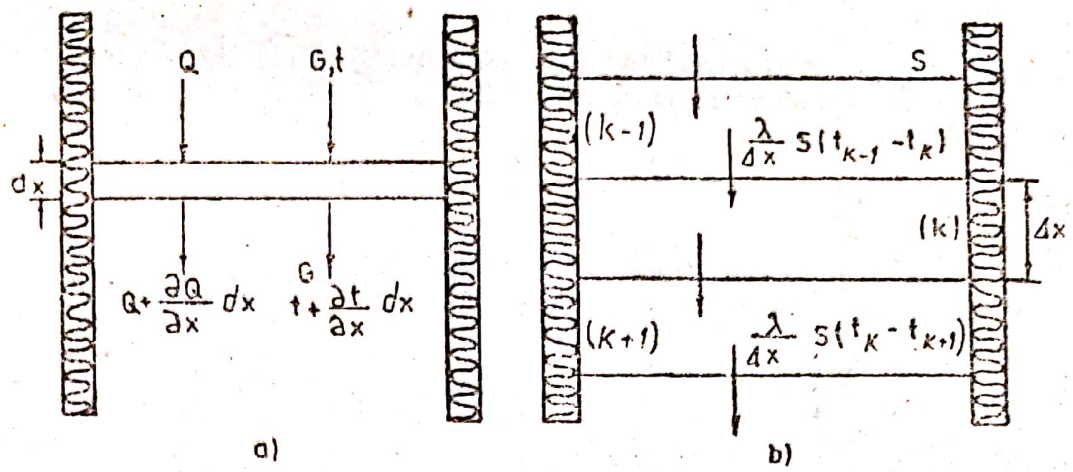


Fig. 3.3. Schema fluxurilor termice
a) metoda caracteristicilor; b) metoda calupurilor

Înlocuind (3.65), (3.66), (3.67) în (3.64) rezultă ecuația cu derivate parțiale:

$$\frac{\partial t}{\partial \tau}+w\frac{\partial t}{\partial x}=\frac{\lambda}{\rho c}\frac{\partial^2 t}{\partial x^2} \tag{3.68}$$

cu condițiile:

$$t(x,\tau)\,|_{x=0}=t_{in}(\tau) \tag{3.69}$$

$$\left.\frac{\partial t}{\partial x}\right|_{x=H}=0 \tag{3.70}$$

$$t(x,\tau)\,|_{\tau=0}=t_0(x) \tag{3.71}$$

Pentru a răspunde unor multiple variante posibile se transformă variabilele și funcția din ecuația (3.68) în variabile și funcție adimensională.

Introducînd notațiile:

$$y=\frac{x}{H},\quad F_0=\frac{a\tau}{H^2};\quad Pe=\frac{wH}{a};\quad \theta=\frac{t-t_0}{t_{in}-t_0} \tag{3.72}$$

unde t_{in} este o temperatură de referință a apei de intrare în rezervor, ecuația cu derivate parțiale (3.68) devine:

$$\frac{\partial \theta}{F_0}+Pe\frac{\partial \theta}{\partial y}=\frac{\partial^2 \theta}{\partial y^2} \tag{3.73}$$

cu condițiile:

$$\theta(0,F_0)=\theta_{in}(F_0) \quad (3.74) \qquad \frac{\partial \theta}{\partial y}(1,F_0)=0 \quad (3.75) \qquad \theta(y,0)=0 \quad (3.76)$$

Pentru rezolvarea modelului matematic definit prin relațiile (3.73)... (3.76) se utilizează metoda caracteristicilor alegînd o rețea spațio-temporală (y-F_0) în care $\Delta y=Pe\cdot\Delta F_0$.

Exprimînd în cadrul acestei relații derivata parțială de ordinul doi prin diferențe centrale, ecuația (3.73) se transformă în sistemul de ecuații diferențiale liniare:

$$\frac{d\theta_0}{dF_0}=-2n^2\theta_0+2n^2\theta_1$$

$$\frac{d\theta_k}{dF_0}=-2n^2\theta_k+n^2\theta_{k-1}+n^2\theta_{k+1} \qquad (k=1,\ 2,\ldots,n-1), \tag{3.77}$$

care trebuie integrat în lungul caracteristicilor

$$\frac{dy}{dF_0}=Pe \tag{3.78}$$

Scanned with OKEN Scanner

Integrînd separat fiecare ecuaţie a sistemului (3.77) pe elemente finite mici ΔF_0 şi ţinînd seama de condiţia la limită (3.75) sub forma $\theta_{n-1}=\theta_{n+1}$ rezultă sistemul algebric:

$$\theta_{1,j+1}+(F-1)\theta_{2,j+1}=E\theta_{0,j}+(F-E)\theta_{1,j}$$

.....

$$\frac{1}{2}(F-1)\theta_{k,j+1}+\theta_{k+1,j+1}+\frac{1}{2}(F-1)\theta_{k+2,j+1}=\frac{1}{2}(F-E)\theta_{k-1,j}+E\,\theta_{k,j}+$$

$$+\frac{1}{2}(F-E)\theta_{k+1,j} \qquad (k=1,2,\ldots,\ n-2) \tag{3.79}$$

.....

$$F(-1)\theta_{n-1,j+1}+\theta_{n,j+1}=\frac{1}{2}(F-E)\theta_{n-2,j}+E\theta_{n-1,j}+\frac{1}{2}(F-E)\theta_{n,j}$$

în care:

$$E=\exp(-2n^2\cdot\Delta F_0); \qquad F=\frac{1-E}{-\ln E}; \qquad \Delta F_0=\frac{1}{n\,\mathrm{Pe}} \tag{3.80}$$

Condiţia la limită (3.74) stabileşte funcţia $\theta_0(F_0)$ de intrare.

Rezolvarea sistemului (3.79) se efectuează matriceal cu relaţia de recurenţă:

$$\Theta_{j+1}=A\,\Theta_j+B \tag{3.81}$$

în care: Θ_{j+1} reprezintă vectorul $\{\theta_{1,j+1}\ \ \theta_{2,j+1}\ldots\theta_{n,j+1}\}$; Θ_j — vectorul $\{\theta_{1,j}\ \ \theta_{2,j}\ldots\theta_{n,j}\}$; A, B — matrice de coeficienţi.

b) O altă tehnică de modelare matematică a fenomenelor de transfer termic din mediul lichid în mişcare este *metoda „calupurilor"*.

Se compartimentează rezervorul în n straturi fiecare din ele fiind la un anumit moment ocupat de către un anumit „calup" de lichid. Timpul de staţionare al calupurilor este:

$$\Delta\tau=\frac{\Delta x}{w}=\frac{1}{n}\frac{H}{w}, \tag{3.82}$$

în care interacţionează termic cu calupurile de lichid din straturile vecine. După intervalul $\Delta\tau$ de timp se efectuează o deplasare instantanee a calupurilor în straturile imediat următoare (fig. 3.3-b).

Bilanţul primului element:

$$\frac{\lambda}{\Delta x}S(t_2-t_1)=S\Delta x\rho c\frac{dt_1}{d\tau} \tag{3.83}$$

conduce la ecuaţia diferenţială:

$$\frac{dt_1}{d\tau}=-\frac{a}{\Delta x^2}t_1+\frac{a}{\Delta x^2}t_2 \tag{3.84}$$

Bilanţul elementului de cîmp conduce la ecuaţia diferenţială:

$$\frac{dt_k}{d\tau}=-2\frac{a}{\Delta x^2}t_k+\frac{a}{\Delta x^2}t_{k-1}+\frac{a}{\Delta x^2}t_{k+1} \qquad (k=2,\ldots,n-1) \tag{3.85}$$

iar bilanţul ultimului element conduce la:

$$\frac{dt_n}{d\tau}=-\frac{a}{\Delta x^2}t_n+\frac{a}{\Delta x^2}t_{n-1}, \tag{3.86}$$

Scanned with OKEN Scanner

rezultînd sistemul (3.84), (3.85) și (3.86), care apelînd la variabilele și funcțiile adimensionale menționate anterior, se transformă în:

$$\frac{d\theta_1}{dF_0} = -n^2\theta_1 + n^2\theta_2$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\frac{d\theta_k}{dF_0} = -2n^2\theta_k + n^2\theta_{k-1} + n^2\theta_{k+1} \qquad (k=2,\ldots,n-1) \tag{3.87}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\frac{d\theta_n}{dF_0} = -n^2\theta_n + n^2\theta_{n-1}$$

Sistemul (3.87) simulează procesul de transfer termic conductiv pe durate scurte ΔF_0, la trecerea de la un interval la altul trebuind efectuate translațiile calupurilor, ceea ce practic înseamnă reinițializarea temperaturilor straturilor cu temperaturile rezultate în straturile imediat anterioare.

Integrînd sistemul (3.87) se obține sistemul algebric:

$$\theta_{1,j+1} + (F^* - 1)\theta_{2,j+1} = E^*\theta_{1,j} + (F^* - E^*)\theta_{2,j}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\frac{1}{2}(F-1)\theta_{k-1,j+1} + \theta_{k,j+1} + \frac{1}{2}(F-1)\theta_{k+1,j+1} = \frac{1}{2}(F-E)\theta_{k-1,j} + E\theta_{k,j} +$$

$$+\frac{1}{2}(F-E)\theta_{k+1,j} \qquad (k=2,3,\ldots,n-1) \tag{3.88}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$(F^* - 1)\theta_{n-1,j+1} + \theta_{n,j+1} = (F^* - E^*)\theta_{n-1,j} + E^*\theta_{n,j}$$

în care:

$$E^* = \exp(-n^2 \cdot \Delta F_0^*); \quad F^* = (1-E^*)/(-\ln E^*); \quad E = \exp(-2n^2 \cdot \Delta F_0^*);$$

$$F = (1-E)/(-\ln E); \qquad \Delta F_0^* = 1/(10n\,\mathrm{Pe}) \tag{3.89}$$

Rezolvarea sistemului (3.89) se face metriceal, folosind relația de recurență (3.81).

3.6. REZOLVAREA CU METODA DIFERENȚELOR FINITE A PROBLEMEI CONDUCȚIEI UNIDIMENSIONALE A CĂLDURII ÎN REGIM TRANZITORIU

Transmiterea căldurii este procesul natural de transfer al energiei interne de la mediile cu temperatură mai mare la cele cu temperatură mai mică sau în interiorul unui mediu de la părți ale acestuia cu temperatură mai scăzută. Căldura se transmite prin *conducție*, *convecție* și *radiație*.

Conducția căldurii este procesul de transmitere a căldurii în interiorul unui corp sau între corpuri aflate în contact direct, de la particulă la particulă, adică între atomi și molecule.

Convecția căldurii este procesul de transmitere a căldurii dintr-o parte a spațiului în alta prin intermediul unui fluid (agent termic) și se produce numai în cazul lichidelor și gazelor.

Scanned with OKEN Scanner

Radiația este procesul de transmitere a căldurii sub formă de energie radiantă. La corpul radiant energia internă se transformă în energie radiantă și se transmite sub formă de unde electromagnetice spre corpul absorbant, unde energia radiantă se transformă în energie internă.

Temperatura t dintr-un corp este o funcție de coordonatele spațiului x, y, z și de timpul τ, de forma:

$$t=f(x,\ y,\ z,\ \tau) \tag{3.90}$$

Această ecuație exprimă matematic cîmpul de temperatură, adică totalitatea temperaturilor din interiorul unui corp. Dacă temperatura nu variază în timp, cîmpul de temperatură se numește staționar, iar în caz contrar se numește cîmp tranzitoriu. După cum temperatura variază în funcție de toate cele trei coordonate ale spațiului, de două din ele sau numai de una, cîmpul se numește respectiv, tri, bi și unidimensional.

Deși fenomenele termice se desfășoară în corpuri tridimensionale, cîmpurile termice rezultate prezintă uneori variații preponderente pe anumite direcții. În astfel de cazuri se pot folosi modele matematice bidimensionale sau unidimensionale, conducînd la simplificarea problemei și la obținerea unei soluții compatibile cu scopul propus.

Modelul unidimensional al conducției căldurii în regim tranzitoriu este descris de ecuația lui Fourier cu derivate parțiale de tip parabolic, de forma:

$$\rho(x,\tau)\,c(x,\tau)\frac{\partial t}{\partial \tau}-\frac{\partial}{\partial x}\left[\lambda(x,\tau)\frac{\partial t}{\partial x}\right]=f(x,\ \tau) \tag{3.91}$$

în care: $\rho(x,\ \tau)$, $c(x,\ \tau)$, $\lambda(x,\ \tau)$ sînt densitatea, căldura specifică și conductivitatea termică a materialului; $t(x,\ \tau)$ — temperatura în punctul de abscisă x și la momentul τ [soluția ecuației (3.91)].

Domeniul pe care se consideră problema este $D=\{0\leqslant x\leqslant 1,\ 0\leqslant \tau\leqslant T,\}$ iar soluția ecuației (3.91) este determinată de condițiile la limită neliniare de forma:

$$\lambda(0,\tau)\frac{\partial t}{\partial x}\bigg|_{x=0}=\varphi_1(\tau,\ t);\qquad -\lambda(1,\ \tau)\frac{\partial t}{\partial x}\bigg|_{x=1}=\varphi_2(\tau,\ t);$$

$$t(x,\ 0)=t_0(x) \tag{3.92}$$

Presupunînd că mediul prin care se transmite căldura este omogen (λ nu depinde de x), ecuația diferențială și condițiile la limită devin:

$$\rho(x,\ \tau)\ c(x,\ \tau)\frac{\partial t}{\partial \tau}-\lambda(\tau)\frac{\partial^2 t}{\partial x^2}=f(x,\ \tau) \tag{3.93}$$

$$\lambda(\tau)\frac{\partial t}{\partial x}\bigg|_{x=0}=\varphi_1(\tau,\ t);\qquad -\lambda(\tau)\frac{\partial t}{\partial x}\bigg|_{x=1}=\varphi_2(\tau,\ t)$$

$$t(x,0)=t_0(x) \tag{3.94}$$

Se aproximează derivatele parțiale cu diferențe finite de forma:

$$\frac{\partial^2 t}{\partial x^2}=\frac{t(x+h_1,\ \tau)-2t(x,\ \tau)+t(x-h_1,\ \tau)}{h}$$

$$\frac{\partial t}{\partial x}=\begin{cases}\dfrac{t(x+h,\tau)-t(x,\ \tau)}{h}, & \text{pentru } x=0\\[2ex] \dfrac{t(x,\ \tau)-t(x-h,\ \tau)}{h}, & \text{pentru } x=1\end{cases} \tag{3.95}$$

$$\frac{\partial t}{\partial \tau}=\frac{t(x,\ \tau)-t(x,\ \tau-e)}{e}$$

Scanned with OKEN Scanner

Dacă se consideră în domeniul D rețeaua rectangulară (fig. 3.4) definită de punctele (x_i, t_j), unde $i=1,\ldots,n-1$; $j=1,\ldots,m$ și $h=1/n$, $e=T/m$, se obțin ecuațiile cu diferențe:

$$-\lambda_j t_{i-1,j}+\left(2\lambda_j+\frac{h^2}{e}r_{ij}\right)t_{ij}-\lambda_j t_{i+1,j}-$$
$$-\frac{h^2}{e}r_{ij}t_{i,j-1}-f_{ij}=0 \qquad (3.96)$$
$$\lambda_j(t_{0j}-t_{1j})+h\varphi_1(\tau_j, t_{0j})=0$$
$$\lambda_j(-t_{n-1,j}+t_{nj})+h\varphi_2(\tau_j,t_{nj})=0$$

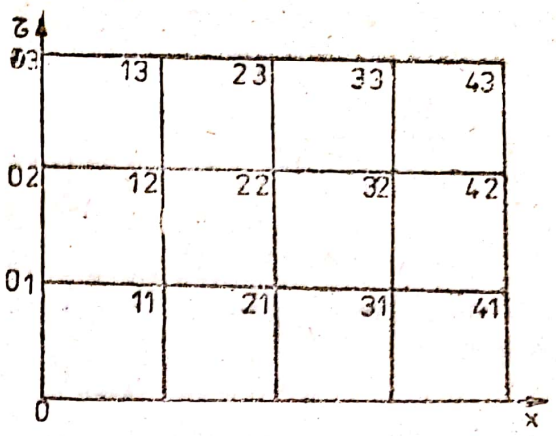


Fig. 3.4. Rețea rectangulară

în care s-au utilizat notațiile: $t_{ij}=t(x_i, \tau_j)$, $r_{ij}=\rho(x_i, \tau_j)\cdot c(x_i, \tau_j)$ $\lambda_j=\lambda(\tau_j)$, $f_{ij}=f(x_i, \tau_j)$.

Din punct de vedere geometric, fiecare punct interior (punct care nu se află pe una din dreptele $x=0$, $x=1$ sau $\tau=0$) generează cîte o relație liniară între valorile lui t în punctul respectiv și în cele trei puncte marcate, iar fiecare punct situat pe dreapta $x=0$ (sau $x=1$) generează cîte o relație neliniară între valorile lui t în punctul respectiv și în punctul din dreapta (sau stînga). Valorile lui t pe dreapta $\tau=0$ sînt cunoscute.

Ecuațiile (3.96) se pot scrie compact sub forma:

$$AX+\Phi(X)=0, \qquad (3.97)$$

în care: A este o matrice de dimensiuni $p\times p$, $p=nm$; X — vector de componente $t_{01},\ldots,t_{n1},\ldots,t_{0m},\ldots,t_{nm}$.

Matricea A se poate scrie sub forma $A=B+C$, unde:

$$B=\begin{bmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_m \end{bmatrix} \qquad (3.98)$$

$$B_j=\begin{bmatrix} \lambda_j & -\lambda_j & & 0 \\ 2\lambda_j+h^2r_{1j}/e & \ddots & \ddots & \\ -\lambda_j & \ddots & \ddots & -\lambda_j \\ & \ddots & \ddots & 2\lambda_j+h^2r_{n-1,j}/e \\ 0 & & -\lambda_j & \lambda_j \end{bmatrix}$$

$$C=\begin{bmatrix} 0 & & & 0 \\ & C_2 & & \\ & & \ddots & \\ 0 & & C_m & 0 \end{bmatrix}, \quad C_j=\begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & -h^2r_{1j}/e & & \\ \vdots & \ddots & \ddots & \vdots \\ & & -h^2r_{n-1,j}/e & 0 \\ 0 & & 0 & 0 \end{bmatrix} \qquad (3.99)$$

Funcția Φ conține valorile funcțiilor φ_1, φ_2 și valorile lui t de pe dreapta $\tau=0$ și are expresia:

$$\Phi(X)=\begin{Bmatrix} f_1(X) \\ f_m(X) \end{Bmatrix} \qquad (3.100)$$

$$f_1(X)=\begin{bmatrix} h\varphi_1(\tau_1, t_{01}) \\ \dfrac{-h^2 r_{11} t_{10}}{e} - f_{11} \\ \vdots \\ \dfrac{-h^2 r_{n-1,1} t_{n-1,0}}{e} - f_{n-1,1} \\ h\varphi_2(\tau_1, t_{n_1}) \end{bmatrix} \qquad f_j(X)=\begin{bmatrix} h\varphi_1(\tau_j, t_{0j}) \\ -f_{1j} \\ \vdots \\ -f_{n-1,j} \\ h\varphi_2(\tau_j, t_{nj}) \end{bmatrix}; \quad j=\overline{2, m}$$

Rezolvarea numerică a ecuației (3.97) se face cu procedeul iterativ simplu:

$$AX^{(k+1)}=-\Phi(X^{(k)}) \tag{3.101}$$

care comportă numai calcularea valorii funcției Φ în punctul curent si rezolvarea unui sistem liniar la fiecare pas de iterație. Componentele soluției ecuației (3.97) reprezintă valori aproximative ale soluției problemei (3.93), (3.94) în punctele rețelei.

3.7. MODELAREA MATEMATICĂ A PROCESULUI DE ABSORBȚIE DIN INSTALAȚIILE FRIGORIFICE

Principiul de funcționare al instalațiilor frigorifice cu absorbție se bazează pe procesul de absorbție a vaporilor de agent frigorific de către mediul absorbant aflat în stare lichidă (soluție săracă), care se produce în aparatul termic denumit absorbitor, luînd naștere soluția bogată ce se introduce în fierbător.

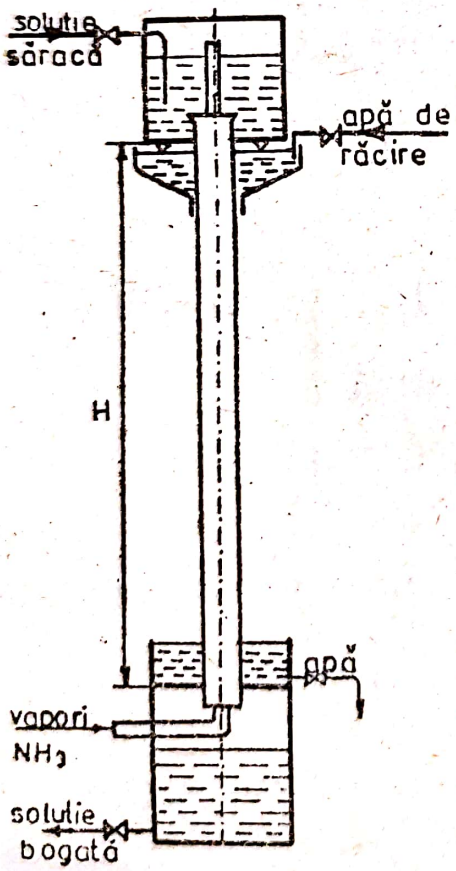


Fig. 3.5. Schema absorbitorului pelicular vertical

Procesul din absorbitor este foarte complex, deoarece se suprapun fenomenele de curgere cu cele de schimb de căldură și de masă. În cazul absorbitoarelor de tip pelicular (fig. 3.5), căldura rezultată în procesul de schimb de masă legată de diferența de temperatură dintre fazele aflate în contact sau din existența surselor interne de căldură (cădură de dizolvare) poate fi mai ușor îndepărtată cu ajutorul apei de răcire.

Se vor stabili ecuațiile diferențiale ce descriu procesul de absorbție al vaporilor în soluție hidroamoniacală în cazul absorbitorului vertical cu pelicula de soluție la interiorul conductei și curgerea în contracurent a vaporilor de amoniac, răcirea realizîndu-se mixt cu aer și apă.

Se fac următoarele ipoteze simplificatoare:

— curgerea soluției în peliculă este unidimensională pe direcția x ($w_y=w_z=0$);

— condițiile nu variază pe direcția z și toate derivatele în raport cu z sînt nule;

— regim staționar de funcționare:

$$\frac{\partial w_x}{\partial \tau}=0; \qquad \frac{\partial \xi}{\partial \tau}=0; \qquad \frac{\partial t}{\partial \tau}=0 \tag{3.102}$$

Scanned with OKEN Scanner

în care: w_x este componenta pe direcția x a vitezei de curgere a soluției în peliculă; ξ — concentrația soluției; t — temperatura soluției;

— difuzia pe direcția x este neglijabilă în raport cu transportul convectiv $(\partial^2\xi/\partial x^2 = 0)$;

— parametrii fizico-chimici ai fazelor aflate în contact se modifică în lungul conductei;

— la interfața lichid-vapori există condiții de echilibru fizico-chimic;

— se cunosc parametrii de intrare (debite, temperaturi, concentrații).

Cu aceste simplificări, ecuațiile de mișcare Navier-Stokes pentru fluide vîscoase și incompresibile, ecuația căldurii și ecuația de continuitate a transferului de masă devin:

$$\eta\frac{\partial^2 w_x}{\partial y^2}+\rho g=0 \tag{3.103}$$

$$w_x\frac{\partial t}{\partial x}=a\frac{\partial^2 t}{\partial y^2} \tag{3.104}$$

$$w_x\frac{\partial \xi}{\partial x}=D\frac{\partial^2 \xi}{\partial y^2} \tag{3.105}$$

unde:

$$a=\frac{\lambda}{\rho c};\qquad D=D_0\left[1+0{,}2\frac{\sqrt{\eta}}{\sqrt[3]{\rho g}}(t-20)\right] \tag{3.106}$$

în care: η, ρ, c. λ sînt vîscozitatea dinamică, densitatea, căldura specifică la presiune constantă și conductivitatea termică a soluției hidroamoniacale; a — difuzivitatea termică; D — coeficientul de difuzie; D_{20} — coeficientul de difuzie la 20°C; g — accelerația gravitațională.

În timpul procesului de absorbție, căldura de absorbție duce la încălzirea soluției din peliculă.

Astfel, la interfața lichid-vapori se poate scrie ecuația de bilanț (3.107) în ipoteza că toată căldura de absorbție pătrunde transversal în peliculă:

$$\lambda\frac{\partial t}{\partial y}\bigg|_{y=\delta}=q_a\rho D\frac{\partial \xi}{\partial y}\bigg|_{y=\delta} \tag{3.107}$$

în care q_a este căldura masică de absorbție.

Din condiția de echilibru la interfața lichid-gaz rezultă legătura între temperatură și concentrație:

$$\xi_i=b_0+bt_i \tag{3.108}$$

în care b_0 și b sînt constante de echilibru determinînd presiunile.

Pentru soluționarea sistemului de ecuații (3.103), (3.104) și (3.105) se aplică metoda aproximativă, constînd în rezolvarea separată mai întîi a ecuației de mișcare (3.103) și apoi utilizînd profilul de viteză obținut se rezolvă numeric cu ajutorul diferențelor finite, ecuațiile pentru transferul termic (3.104) și de masă (3.105).

Ecuația (3.103) se integrează cu condițiile pe frontieră:

$$\begin{aligned} w_x&=0, \text{ pentru } y=0\\ \partial w_x/\partial y&=0, \text{ pentru } y=\delta, \end{aligned} \tag{3.109}$$

obținîndu-se pentru viteza locală expresia:

$$w_x=w_M\left[1-\left(\frac{y-\delta}{\delta}\right)^2\right] \tag{3·110}$$

unde:

$$w_M=\frac{\rho g\delta^2}{2\eta} \tag{3.111}$$

Scanned with OKEN Scanner

în care: δ este grosimea peliculei de soluție; w_M — viteza maximă a soluției în peliculă.

Trebuie determinate în continuare variațiile de temperatură și de concentrație atît în grosimea peliculei (δ) cît și pe lungimea ei (H), respectiv ξ și t pe domeniul $[0, H] \times [0, \delta]$, prin rezolvarea sistemului (3.104), (3.105), ținînd seama de:

— condițiile inițiale:

$$t(0, y) = t_0; \quad \xi(0.y) = \xi_0 \tag{3.112}$$

— condițiile la limită (la perete):

$$t(0,0) = t_0; \quad t(x, 0) = t_p(x); \quad \xi(x,0) = \xi_p \tag{3.113}$$

— condițiile la interfața lichid-gaz:

$$\xi(x,\delta) = b_0 + b\, t(x, \delta) \tag{3.114}$$

$$\lambda \frac{\partial t(x, y)}{\partial y}\bigg|_{y=\delta} = q_a \rho D \frac{\partial \xi(x, y)}{\partial y}\bigg|_{y=\delta}$$

în care parametrii D, λ, ρ, q_a sînt funcție de ξ și t, astfel că ecuațiile sînt puternic neliniare și pentru rezolvare se aplică metoda numerică a diferențelor finite.

Pelicula se divizează într-o rețea rectangulară (fig. 3.6) în care derivatele parțiale se înlocuiesc cu diferențe finite:

$$\frac{\partial t}{\partial x} = \frac{t_{i,j} - t_{i-1,j}}{h}$$
$$\frac{\partial \xi}{\partial x} = \frac{\xi_{i,j} - \xi_{i-1,j}}{h} \tag{3.115}$$

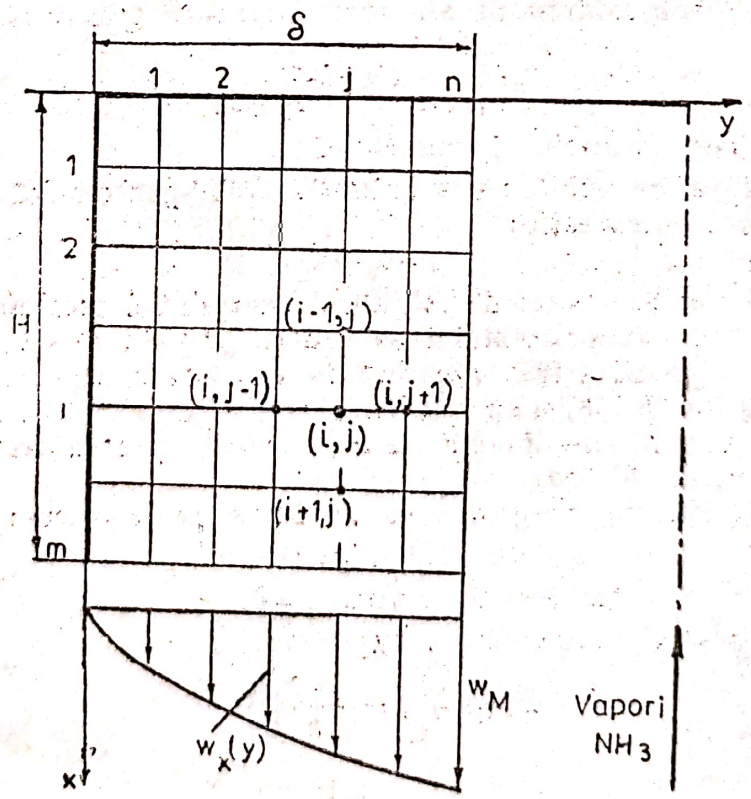


Fig. 3.6. Discretizarea peliculei de soluție

$$\frac{\partial^2 t}{\partial y^2}=\frac{t_{i,j+1}-2t_{i,j}+t_{i,j-1}}{e^2}$$
$$\frac{\partial^2 \xi}{\partial y^2}=\frac{\xi_{i,j+1}-2\xi_{i,j}+\xi_{i,j-1}}{e^2} \qquad (3.116)$$

unde $h=H/m$ și $e=\delta/n$ sînt pașii pe direcția x și respectiv y.

Ecuațiile (3.104), (3.105) devin:

$$-t_{i,j-1}+\left(2+\frac{1}{u_{i,j}}\right)t_{i,j}-t_{i,j+1}=\frac{1}{u_{i,j}}t_{i-1,j}$$
$$-\xi_{i,j-1}+\left(2+\frac{1}{v_{i,j}}\right)\xi_{i,j}-\xi_{i,j+1}=\frac{1}{u_{i,j}}\xi_{i-1,j} \qquad (3.117)$$
$$(j=2,3,\ldots\ldots,n-1)$$

în care:

$$\frac{1}{u_{i,j}}=\frac{e^2 w_{i,j}}{a(\xi_{i-1,j},\, t_{i-1,j})h}$$
$$\frac{1}{v_{i,j}}=\frac{e^2 w_{i,j}}{D(\xi_{i-1,j},\, t_{i-1,j})h} \qquad (3.118)$$

iar condițiile de integrare (3.112), (3.113), (3.114) primesc forma:

$$t_{1,j}=t_0; \quad \xi_{1,j}=\xi_0; \quad t_{i,1}=t_p(x); \quad \xi_{i,1}=\xi_p \qquad (3.119)$$

$$t_{i,n-1}=t_{i,n}-\frac{q_a(\xi_{i,n},t_{i,n})\cdot\rho(\xi_{i,n},t_{i,n})\cdot D(\xi_{i,n},t_{i,n})}{\lambda(\xi_{i,n},\, t_{i,n})}(\xi_{i,n}-\xi_{i,n-1}) \qquad (3.120)$$

$$\xi_{i,n}=b_0+b\,t_{i,n} \qquad (3.121)$$

Funcțiile $\eta(\xi, t)$, $a(\xi, t)$, $D(\xi, t)$, $\rho(\xi, t)$, $\lambda(\xi, t)$, $q_a(\xi, t)$ se iau din literatura de specialitate fie sub formă analitică, fie tabelară sau de diagrame, iar grosimea peliculei δ și viteza w_M sînt date cunoscute.

Sistemul (3.117) se rezolvă separat în t și separat în ξ.

Cunoscînd variațiile de temperatură și de concentrație atît în grosimea peliculei cît și în lungul absorbitului, determinate prin metoda cu diferențe finite, se poate determina pe baza teoriei filmului:

— fluxul unitar termic:

$$q=-\lambda\frac{\partial t}{\partial y}\bigg|_{y=0} \qquad (3.122)$$

sau:

$$q=\alpha(t_i-t_{p1})=\frac{\lambda_c}{\delta_c}(t_{p1}-t_{p2})=\alpha_W(t_{p2}-t_W) \qquad (3.123)$$

în care: λ este conductivitatea termică a soluției; t_i — temperatura la interfața lichid-vapori; t_{p1} — temperatura la peretele interior al conductei; t_{p2} — temperatura la peretele exterior al conductei; t_W — temperatura în pelicula de apă; λ_c, $\delta_c=(d_e-d_i)/2$ — conductivitatea termică și grosimea conductei; α_W — coeficientul de transfer termic convectiv în pelicula de apă;

— fluxul unitar de masă:

$$q_*=-D\frac{\partial \xi}{\partial y}\bigg|_{y=\delta} \qquad (3.124)$$

Avînd aceste valori, se obțin coeficienții de transfer convectiv și de transfer de masă β în pelicula de soluție, cu relațiile:

$$\alpha(x)=\frac{q}{t_i-t_{p1}} \qquad (3.125) \qquad \beta(x)=\frac{q_s}{\rho_s\xi_s-\rho\xi} \qquad (3.126)$$

unde ρ_s și ξ_s sînt densitatea și concentrația soluției la saturație.

Rezultă apoi valorile medii α_m și β_m, definite prin relațiile:

$$\alpha_m=\frac{1}{H}\int\alpha(x)\mathrm{d}x \qquad (3.127) \qquad \beta_m=\frac{1}{H}\int\beta(x)\,\mathrm{d}x, \qquad (3.128)$$

precum și valorile medii ale fluxului unitar termic și ale fluxului unitar de masă:

$$q_m=\frac{1}{H}\int q(x)\,\mathrm{d}x \qquad (3.129) \qquad q_{sm}=\frac{1}{H}\int q_s(x)\,\mathrm{d}x \qquad (3.130)$$

3.8. ANALIZA NUMERICĂ CU ELEMENTE FINITE A PROBLEMELOR DE CONDUCȚIE A CĂLDURII

Transmiterea căldurii și în special conducția termică constituie un domeniu în care metoda elementelor finite a găsit un cîmp larg de aplicare.

3.8.1. PROPRIETĂȚI CARACTERISTICE ELEMENTELOR FINITE NODALE

3.8.1.1. Proprietăți topologice

Metoda elementelor finite poate fi descrisă ca un procedeu sistematic prin care orice funcție continuă este aproximată printr-un model discret format dintr-un set de valori ale funcției date la un număr finit de puncte alese în domeniul ei, împreună cu aproximări locale pe porțiuni ale funcției peste un număr finit de subdomenii. Aproximările locale ale funcției pe fiecare subdomeniu sînt definite în raport cu valorile discrete ale funcției la punctele selecționate inițial. Astfel, structura unui model cu elemente finite implică:

a) identificarea unui număr finit de puncte în domeniul funcției de aproximat și specificarea valorilor acestei funcții în punctele considerate. Aceste puncte se numesc *noduri*, iar valorile funcției în noduri se numesc *valori nodale*, ce se determină prin calcul (experiment numeric);

b) domeniul funcției este reprezentat ca o colecție finită pe subdomenii interconectate, de dimensiuni finite. Aceste subdomenii se numesc *elemente finite*. Modelul domeniului actual apare deci ca un ansamblu al acestor elemente finite conectate împreună la noduri, pe frontierele lor comune;

c) funcția dată este aproximată local, pe fiecare element, prin funcții continue care sînt definite unic în raport cu valorile funcției date (sau a funcției și a valorilor derivatelor ei pînă la un anumit ordin) la nodurile ce aparțin fiecărui element.

Pentru o reprezentare mai formală a ideilor de bază se consideră cazul unei probleme unidimensionale. Fie funcția continuă $F: D\to R$, unde domeniul $D=[A, B]$ și funcția $\Phi: \bar{D}\to R$, un model de aproximare cu elemente finite a funcției date, unde $\bar{D}\subseteq D$ (fig. 3.7).

Scanned with OKEN Scanner

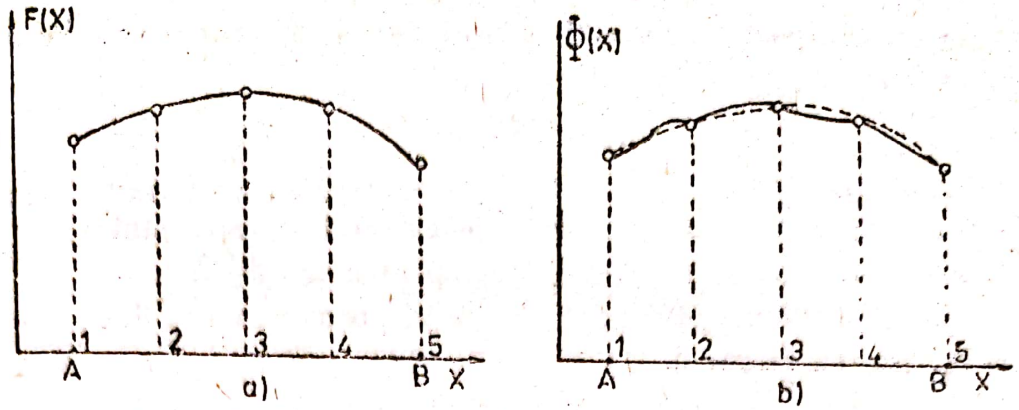


Fig. 3.7. Graficul funcției date $F(X)$ și al funcției de aproximare $\Phi(X)$

Construcția modelului cu elemente finite nodale se face astfel:

— se selectează un număr finit de noduri $N_n \in N$ având coordonatele globale $X_i \in D(i=1,2,\ldots,N_n)$. Valorile globale ale funcției în aceste puncte $F_i=F(X_i)$ se denumesc valori nodale. Spre exemplificare, se alege un număr de cinci noduri ($N_n=5$) și se consideră că pentru toate aceste noduri $F_i=\Phi_i$ (fig. 3.8 — *a*);

— se selectează un număr finit de elemente $N_e \in N$, fiecărui element asociindu-i un anumit număr de noduri $N_n^e \in N$. Pentru exemplul considerat se aleg patru elemente, fiecare element având cîte două noduri ($N_e=4$, $N_n^e=2$), (fig. 3.8 — b) ;

— fiecare element e ($e=1,\ldots,4$) se caracterizează acum, în mod local, prin coordonatele nodurilor sale x_j^e și valorile nodale ale funcției de aproximare f_j^e, pentru $j=1,2$;

— se stabilesc relații de discretizare a domeniului de aproximare în elemente finite, de forma (fig. 3.8):

$$x_1^1=X_1,\ x_2^1=X_2,\ \text{pentru } e=1;\ x_1^2=X_2,\ x_2^2=X_3,\ \text{pentru } e=2;$$
$$x_1^3=X_3,\ x_2^3=X_4,\ \text{pentru } e=3;\ x_1^4=X_4,\ x_2^4=X_5,\ \text{pentru } e=4$$

Aceste relații se pot scrie sub forma generală:

$$x_j^e=\omega_{ji}^e X_i \quad (i=1,\ldots,N_n\ j=1,\ldots,N_n^e;\ e=1,\ldots,N_e) \tag{3.131}$$

în care elementele ω_{ij}^e ale matricei booleene Ω^e sînt definite de relația:

$$\omega_{ji}^e=\begin{cases} 1, & \text{dacă nodul local } j \text{ al elementului } e \text{ este incident pe nodul global } i; \\ 0, & \text{în caz contrar} \end{cases} \tag{3.132}$$

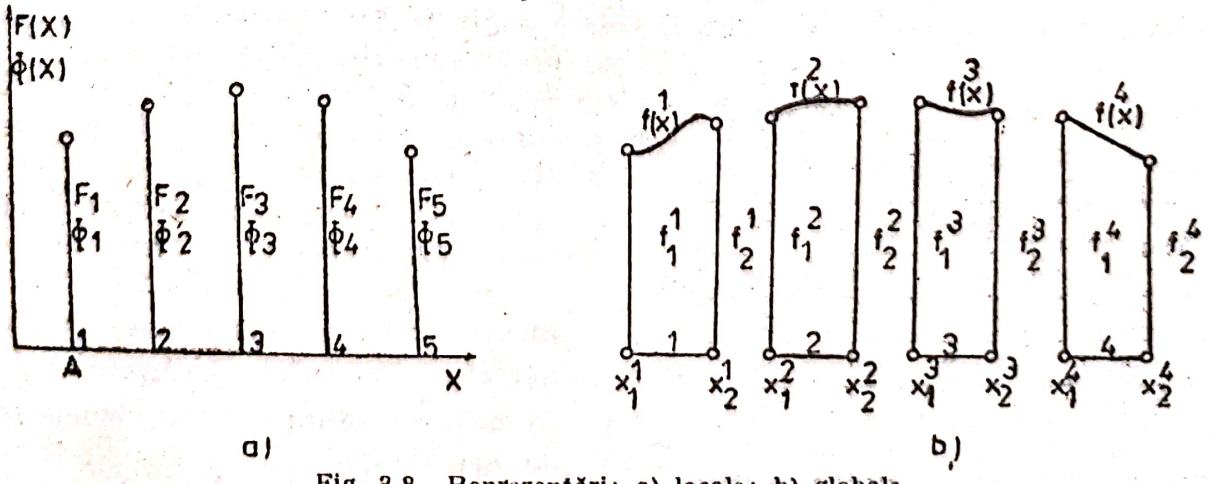


Fig. 3.8. Reprezentări: a) locale; b) globale

Relația (3.131) poate fi folosită în mod similar și pentru valorile nodale ale funcției date:

$$f_j^e = \omega_{ji}^e F_i \tag{3.133}$$

— se stabilesc relații de asamblare ale fiecărui element finit e în domeniul de aproximare $\bar{D}$. Pentru domeniul considerat, acestea sînt:

$X_1 = x_1^1$, pentru nodul 1; $X_2 = x_2^1 = x_1^2$, pentru nodul 2;

$X_3 = x_2^2 = x_1^3$, pentru nodul 3; $X_4 = x_2^3 = x_1^4$, pentru nodul 4;

$X_5 = x_2^4$, pentru nodul 5.

Aceste relații pot fi scrise în mod generic astfel:

$$X_i = \lambda_{ij}^e x_j^e \quad (i=1,\ldots,N_n;\ j=1,\ldots,N_n^e;\ e=1,\ldots,N_e) \tag{3.134}$$

unde λ_{ij}^e are ca rezultat conectarea elementului e la domeniul de aproximare $\bar{D}$. O relație similară poate fi scrisă și pentru funcții:

$$F_i = \lambda_{ij}^e f_j^e \tag{3.135}$$

Se observă că transformările ω_{ji}^e și λ_{ij}^e sînt una inversă celeilalte, putîndu-se scrie:

$$\omega_{ji}^e \lambda_{ij}^e = \delta_{ij} \tag{3.136}$$

unde δ_{ij} este simbolul lui Kronecker.

— relațiile de discretizare împreună cu cele de asamblare se reunesc în aplicațiile practice de programare pe calculator formînd *matrice de conexiuni*. Această matrice stabilește legături de apartenență dintre elemente, nodurile locale și cele globale. Pentru exemplul considerat, matricea de conexiuni este prezentată pe elemente în tabelul 3.1.

Modelul prezentat anterior poate fi generalizat pentru două și trei dimensiuni, păstrînd aceeași logică.

3.8.1.2. Sisteme de coordonate

După cum a rezultat din modelul de aproximare prezentat anterior, punctele nodale sînt identificate prin două sisteme de numerotare: unul global, pentru întregul domeniu de analiză, și altul local, pentru fiecare element în parte. Este convenabil de a se asocia sistemului local de noduri și un *sistem local de coordonate*. Originea acestui sistem poate fi aleasă, practic, oriunde în interiorul unui element. Se preferă totuși alegerea nodală (la unul dintre nodurile elementului) sau alegerea centrului de greutate al elementului (fig. 3.9.).

Tabelul 3.1

Matrice de conexiuni

Elementul	Noduri globale pentru	
	$j=1$	$j=2$
1	1	2
2	2	3
3	3	4
4	4	5

La rîndul lor, coordonatele locale pot fi *normale* (*carteziene*) și *naturale*, denumite și coordonate de lungime în cazul elementelor unidimensionale, coordonate de arie în cazul elementelor bidimensionale și coordonate de volum în cazul elementelor tridimensionale.

Scanned with OKEN Scanner

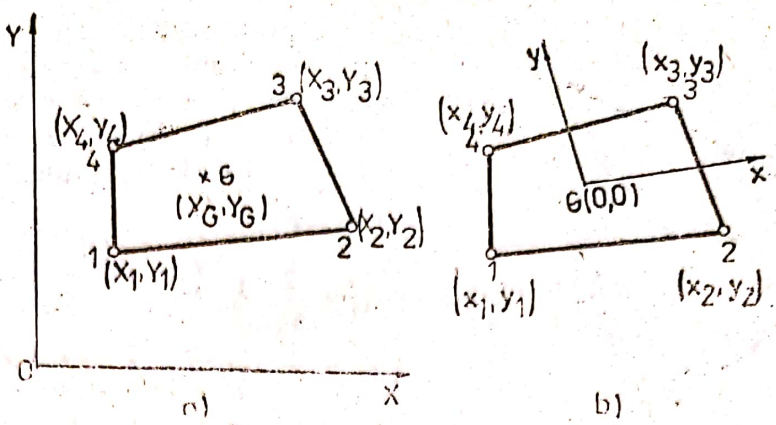


Fig. 3.9. Coordonate carteziene:
a) globale; b) locale

Coordonatele naturale sînt un fel de coordonate normalizate, în sensul că se obțin prin raportarea coordonatelor globale, sau a unor arii determinate de punctul generic, la mărimi caracteristice elementului finit (lungimi, arii). Dacă se alege drept origine pentru aceste coordonate naturale centrul elementului, atunci domeniul lor de variație este $[-1,1]$, iar dacă în loc de punct unic de origine se aleg origini nodurile elementului, atunci domeniul lor de variație este $[0,1]$.

Avantajul coordonatelor naturale constă în special în simplificarea operațiilor de integrare pe cuprinsul elementului finit.

Se prezintă în continuare cele mai folosite sisteme de coordonate naturale.

Fie un element unidimensional de tip liniar (fig. 3.10). Alegînd drept origine a sistemului local de referință centrul de greutate al elementului, se obține sistemul ξ — natural (fig. 3.10, b). Pentru un punct oarecare cu $X \in [X_i, X_j]$ se poate scrie:

$$\xi = \frac{X - X_G}{a}, \quad \xi \in [-1, +1] \tag{3.137}$$

Acest sistem de referință se preferă în cazul folosirii integrării numerice.

Alegînd drept origini chiar nodurile elementului, se obține sistemul L — natural (fig. 3.10, c). Pentru un punct oarecare cu $X \in [X_i, X_j]$ se scrie:

$$X = L_i X_i + L_j X_j \tag{3.138}$$

$$L_i + L_j = 1 \tag{3.139}$$

Prima relație arată caracterul de funcție de pondere al coordonatelor L, iar cea de-a doua relație arată că L_i și L_j nu sînt variabile independente. Relațiile (3.138) și (3.139) conduc la:

$$L_i = \frac{X_j - X}{2a}; \qquad L_j = \frac{X - X_i}{2a} \tag{3.140}$$

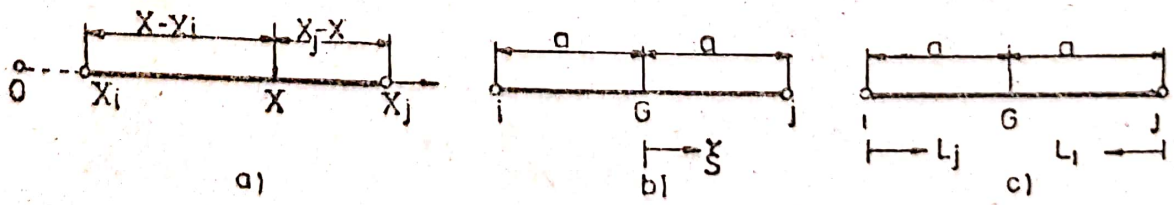


Fig. 3.10. Coordonate pentru elemente unidimensionale:
a) globale; b) ξ-naturale; c) L-naturale

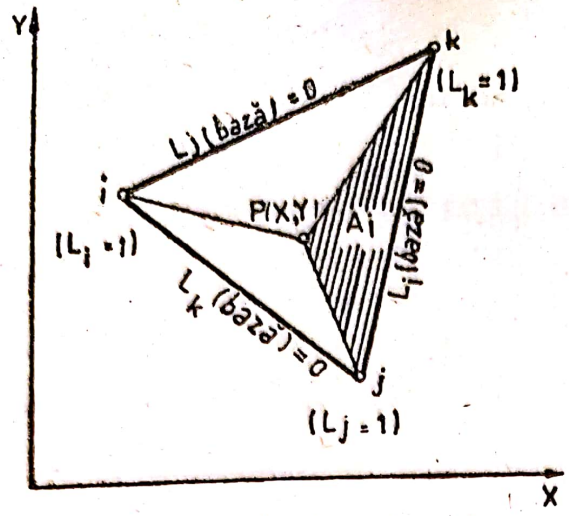


Fig. 3.11. Coordonate L-naturale pentru elemente triunghiulare.

Acest sistem se preferă atunci cînd expresiile integrale caracteristice elementului finit se evaluează direct (analitic). În scopul uşurării evaluării lor se foloseşte formula:

$$\int_{X_i}^{X_j} L_i^{\alpha} L_j^{\beta}\, dx = \frac{\alpha!\beta!(X_j - X_i)}{(\alpha+\beta+1)!} \qquad (3.141)$$

în care α şi β sînt numere întregi.

În figura 3.11 sînt prezentate coordonatele L — naturale pentru elemente finite triunghiulare cu trei noduri.

Coordonatele naturale ale unui punct $P(X, Y)$ de pe elementul finit ijk sînt rapoartele de arie:

$$L_i = \frac{A_i}{A_e}; \qquad L_j = \frac{A_j}{A_e}; \qquad L_k = \frac{A_k}{A_e} \qquad (3.142)$$

în care A_e este aria elementului finit, iar A_i, A_j, A_k sînt porţiuni din această arie determinate de punctul P.

Legătura dintre coordonatele globale şi cele L — naturale este dată de relaţia matriceală:

$$\begin{Bmatrix} X \\ Y \\ 1 \end{Bmatrix} = \begin{bmatrix} X_i & X_j & X_k \\ Y_i & Y_j & Y_k \\ 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} L_i \\ L_j \\ L_k \end{Bmatrix} \qquad (3.143)$$

Se observă din figura 3.11 că domeniul de variaţie al acestor coordonate este [0,1], cu valoare unitară în nodul cu acelaşi număr şi valoarea zero pe latura opusă nodului respectiv. Avantajul folosirii lor constă în posibilitatea unei evaluări mai rapide a expresiilor elementale integrale folosind relaţia:

$$\int_{A_e} L_i^{\alpha} L_j^{\beta} L_k^{\gamma}\, dA = \frac{\alpha!\beta!\gamma!}{(\alpha+\beta+\gamma+2)!} 2A_e \qquad (3.144)$$

în care α, β, γ sînt numere întregi, iar:

$$L_i = \frac{1}{2A_e}(a_i + b_i X + c_i Y)$$

$$L_j = \frac{1}{2A_e}(a_j + b_j X + c_j Y) \qquad (3.145)$$

$$L_k = \frac{1}{2A_e}(a_k + b_k X + c_k Y)$$

$$A_e = \frac{1}{2}\begin{vmatrix} 1 & X_i & Y_i \\ 1 & X_j & Y_k \\ 1 & X_k & Y \end{vmatrix} = \text{aria elementului } ijk \qquad (3.146)$$

$$a_i = X_j Y_k - X_k Y_j; \qquad b_i = Y_j - Y_k; \qquad c_i = X_k - X_j \qquad (3.147)$$

Ceilalţi coeficienţi se obţin prin permutări ciclice ale indicilor.

Scanned with OKEN Scanner

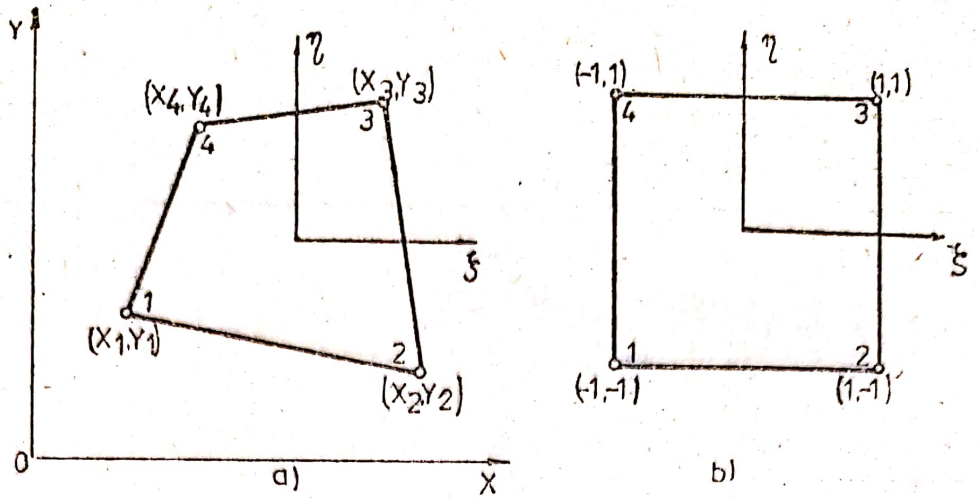


Fig. 3.12. Coordonate pentru elemente patrulatere:
a) globale; b) ξ-naturale

În figura 3.12 se prezintă coordonatele ξ — naturale pentru un patrulater oarecare. Relațiile dintre coordonatele naturale (ξ, η) și cele globale (X, Y) au expresiile:

$$X=\frac{1}{4}[(1-\xi)(1-\eta)X_1+(1+\xi)(1-\eta)X_2+(1+\xi)(1+\eta)X_3+$$
$$+(1-\xi)(1+\eta)X_4] \tag{3.148}$$

$$Y=\frac{1}{4}[(1-\xi)(1-\eta)Y_1+(1+\xi)(1-\eta)Y_2+(1+\xi)(1+\eta)Y_3+$$
$$+(1-\xi)(1+\eta)Y_4] \tag{3.149}$$

3.8.2. INTEGRAREA NUMERICĂ A EXPRESIILOR ELEMENTALE

3.8.2.1. Aplicarea metodei Newton — Côtes

Metoda elementelor finite este în esență o metodă integrală, operînd la nivelul elementelor finite folosite. Expresiile rezultatelor sub semnul integral sînt de cele mai multe ori produse de matrice. Evaluarea lor directă este foarte dificilă, fapt pentru care se apelează frecvent la integrarea numerică.

Integralele de evaluat pot fi scrise sub forma:

$$\int_{-1}^{1} f(\xi)\,d\xi;\quad \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)d\xi\,d\eta;\quad \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)d\xi\,d\eta\,d\zeta$$

unde: f reprezintă de regulă un produs de matrice; ξ, η, ζ — coordonate ale elementului; $[-1,1]$ — domeniul normalizat de variație a coordonatelor.

Evaluarea numerică a acestor integrale se face folosind următoarele relații generale:

$$\int_{-1}^{1} f(\xi)d\xi=\sum_{i=1}^{n} W_i f(\xi_i)$$

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)d\xi\,d\eta=\sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j f(\xi_i,\eta_j) \tag{3.150}$$

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)d\xi\,d\eta\,d\zeta=\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} W_i W_j W_k f(\xi_i,\eta_j,\zeta_k)$$

Scanned with OKEN Scanner

Tabelul 3.2

Valori ale coeficienților Côtes de integrare numerică

$n-1$	$\bar{w}_1$	$\bar{w}_2$	$\bar{w}_3$	$\bar{w}_4$	$\bar{w}_5$	$\bar{w}_6$	$\bar{w}_7$	$\bar{w}_8$	$\bar{w}$	N
0	1	2	3	4	5	6	7	8	9	10
1	1	1								2
2	1	4	1							6
3	1	3	3	1						8
4	7	32	12	32	7					90
5	19	75	50	50	75	19				288
6	41	216	27	272	27	216	41			840
7	751	3577	1323	2989	2989	1329	3577	751		17280
8	989	5888	−928	10496	−4540	10496	−928	5888	989	28350

în care:

W_i, W_j, W_k sînt coeficienți de pondere ce se iau din tabelul 3.2; ξ_i, η_j, ζ_k — puncte de evaluare a integrandului. Limitele de integrare pot fi oarecare, dar dacă se folosesc coordonate naturale ele sînt (−1) și (+1).

Aproximarea funcției în n puncte selectate a priori se face determinînd un polinom de ordinul $n-1$ care trece prin aceste puncte și care este apoi integrat exact. De regulă cele n puncte se poziționează echidistant în intervalul de integrare $[-1,1]$.

Prin particularizare se obține:

— pentru $n=2$, formula trapezului:

$$\int_{-1}^{1} f(\xi)\,d\xi = f(-1)+f(1) \tag{3.151}$$

— pentru $n=3$, formula lui Simpson:

$$\int_{-1}^{1} f(\xi)\,d\xi = \frac{1}{3}[f(-1)+4f(0)+f(1)] \tag{3.152}$$

— pentru $n=4$:

$$\int_{-1}^{1} f(\xi)\,d\xi = \frac{1}{4}\left[f(-1)+3f\left(-\frac{1}{3}\right)+3f\left(\frac{1}{3}\right)+f(1)\right] \tag{3.153}$$

3.8.2.2. Aplicarea metodei Gauss — Legendre

În loc de predeterminarea punctelor de evaluare, această metodă permite determinarea acestora prin calcul, astfel încît eroarea să fie minimă pentru un număr dat de puncte de evaluare. Metoda optimizează distanța dintre puncte, astfel ca pentru același ordin de mărime al erorii să se folosească un număr mai mic decît în cazul metodei Newton-Côtes. Dacă integrarea se face în n puncte Gauss, gradul polinomului trebuie să fie cel mult $2n-1$.

În funcție de numărul n de puncte Gauss de integrare, coeficienții de pondere W_k și coordonatele ξ_k se iau din tabelul 3.3, iar pentru integrare se folosesc direct formulele (3.150).

Scanned with OKEN Scanner

Tabelul 3.3

Elemente ale formulei de integrare numerică Gauss-Legendre

n	k	ξ_k	W_k
0	1	2	3
1	1	0	2
2	1;2	∓0,57735027	1
3	1;3	∓0,77459667	0,55555556
	2	0	0,88888889
4	1;4	∓0,86113631	0,34785484
	2;3	∓0,33998104	0,65214516
5	1;5	∓0,90617985	0,23692688
	2;4	∓0,53846931	0,47862868
	3	0	0,56888889
6	1;6	∓0,93246951	0,17132450
	2;5	∓0,66120939	0,36076158
	3;4	∓0,23861919	0,46791394
7	1;7	∓0,94910791	0,12948496
	2;6	∓0,74153119	0,27970540
	3;5	∓0,40584515	0,38183006
	4	0	0,41795918
8	1;8	∓0,96028986	0,10122854
	2;7	∓0,79666648	0,22238104
	3;6	∓0,52553242	0,31370664
	4;5	∓0,18343464	0,36268378

3.8.3. TIPURI DE ELEMENTE FINITE ȘI FUNCȚII DE INTERPOLARE

Rezolvarea problemelor cu ajutorul metodei elementelor finite începe cu alegerea tipului de element finit, urmînd discretizarea domeniului cu elemente finite.

Evoluția folosirii metodei elementelor finite a fost însoțită de diversificarea tipurilor de elemente, în continuare studiindu-se cele mai folosite dintre acestea. Elementul finit triunghiular cu trei noduri a fost și continuă să fie folosit în analiza cu elemente finite pentru rezolvarea multor probleme. Pentru anumite probleme, elementele finite de ordin inferior ca cel triunghiular nu oferă însă rezultate cu suficientă precizie, recomandîndu-se în aceste cazuri folosirea elementelor finite de ordin superior.

3.8.3.1. Generarea funcțiilor de interpolare

O dată cu alegerea tipului de element finit se aleg și *funcțiile de aproximare*, încît descrierea elementelor finite este însoțită de prezentarea funcțiilor de aproximare asociate.

Funcția de aproximare Φ se poate scrie sub forma:

$$\Phi = \sum_{i=1}^{n} N_i \Phi_i = [N]\{\Phi\}_e \qquad (3.154)$$

în care $\{\Phi\}_e$ este vectorul valorilor nodale ale funcției de aproximare pentru elementul generic e; $[N]$ — matricea funcțiilor de formă a acestui element.

Scanned with OKEN Scanner

Drept funcţii de aproximare se folosesc mai ales polinoame de interpolare datorită uşurinţei de derivare şi de integrare a acestora. De aceea funcţiile de aproximare se mai numesc şi *funcţii de interpolare*. Ordinul polinomului de interpolare depinde de numărul gradelor de libertate asociate elementului finit. Cu alte cuvinte, numărul coeficienţilor polinomiali trebuie să fie egal cu numărul nodurilor disponibile pentru a-i evalua.

De exemplu, pentru un *element triunghiular liniar*, funcţia de aproximare este un polinom de gradul I, de forma:

$$\Phi = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{3.155}$$

Polinomul de interpolare trebuie să fie în acest caz de ordinul întîi, deoarece pe o latură a elementului finit sînt două noduri şi polinomul de gradul întîi interpolează în două puncte. Numărul de coeficienţi α_j, numiţi coordonate generalizate ale elementului, este de trei, deoarece triunghiul are trei noduri şi acestea se determină impunînd ca în noduri funcţia să capete valorile Φ_i, Φ_j, Φ_k:

$$\begin{aligned} \Phi_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 y_i \\ \Phi_j &= \alpha_1 + \alpha_2 x_j + \alpha_3 y_j \\ \Phi_k &= \alpha_1 + \alpha_2 x_k + \alpha_3 y_k \end{aligned} \tag{3.156}$$

Rezolvînd sistemul (3.156) în raport cu coordonatele generalizate α_j, se obţine:

$$\begin{aligned} \alpha_1 &= \frac{1}{2A_e}(a_i\Phi_i + a_j\Phi_j + a_k\Phi_k) \\ \alpha_2 &= \frac{1}{2A_e}(b_i\Phi_i + b_j\Phi_j + b_k\Phi_k) \\ \alpha_3 &= \frac{1}{2A_e}(c_i\Phi_i + c_j\Phi_j + c_k\Phi_k) \end{aligned} \tag{3.157}$$

în care A_e este aria elementului finit:

$$A_e = \frac{1}{2}\begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} \tag{3.158}$$

iar coeficienţii a, b, c au expresiile:

$$\begin{aligned} &a_i = x_j y_k - x_k y_j; \quad && a_j = x_k y_i - x_i y_k; \quad && a_k = x_i y_j - x_j y_i \\ &b_i = y_i - y_k; && b_j = y_k - y_i; && b_k = y_i - y_j \\ &c_i = x_k - x_j; && c_j = x_i - x_k; && c_k = x_j - x_i \end{aligned} \tag{3.159}$$

şi sînt constanţi deoarece depind numai de coordonatele constante ale nodurilor.

Introducînd (3.157) în (3.155) se obţine modelul de aproximare:

$$\Phi = N_i\Phi_i + N_j\Phi_j + N_k\Phi_k = [N_i N_j N_k]\begin{Bmatrix} \Phi_i \\ \Phi_j \\ \Phi_k \end{Bmatrix} \tag{3.160}$$

Scanned with OKEN Scanner

în care funcțiile de formă ale elementului finit triunghiular sînt date de relațiile:

$$N_i=\frac{1}{2A_e}(a_i+b_ix+c_iy); \qquad N_j=\frac{1}{2A_e}(a_j+b_jx+c_jy);$$

$$N_k=\frac{1}{2A_e}(a_k+b_kx+c_ky) \tag{3.161}$$

Relația (3.160) exprimă variația aproximată Φ a funcției necunoscute în raport cu valorile acestei funcții în nodurile elementului finit prin intermediul funcțiilor de formă. Această variație este liniară, funcțiile de formă sînt și ele liniare, iar elementul finit cu trei noduri se numește liniar.

Valoarea funcților de formă variază între 0 și 1. Ele au valoarea 1 în nodul la care se referă și 0 în celelalte noduri. Astfel, $N_i=1$ în nodul i și $N_i=0$ în nodurile j și k. Totodată aceste funcții de formă sînt aceleași ca și coordonatele L — naturale ale elementului finit triunghiular.

În mod similar se poate proceda și pentru celelalte elemente bidimensionale. Polinomul general de aproximare de ordinul n este dat de relația:

$$\Phi=\sum_{k=1}^{T_n}\alpha_k x^i y^j; \qquad i+j\leqslant n \tag{3.162}$$

în care numărul termenilor din polinom este $T_n=(n+1)(n+2)/2$.

În expresiile polinoamelor de interpolare (3.155) și (3.160) funcția necunoscută a fost considerată o mărime scalară. În orice punct al elementului finit există un singur parametru necunoscut, sau un singur grad de libertate. Acesta este cazul, de exemplu, al problemelor de conducție termică, la care există o singură necunoscută, temperatura. În alte probleme, în fiecare nod de element finit, funcția necunoscută poate avea mai multe componente, fiind deci o mărime vectorială. În acest caz problema se rezolvă la fel, dar în raport cu componentele după direcțiile axelor de coordonate ale funcției. Aceste componente se notează cu u după direcția x, cu v, după direcția y și cu w după direcția z.

În cazul elementului finit triunghiular, pentru fiecare componentă u și v există un model de aproximare de același fel cu (3.160):

$$\begin{aligned} u&=N_iu_i+N_ju_j+N_ku_k \\ v&=N_iv_i+N_jv_j+N_kv_k \end{aligned} \tag{3.163}$$

funcțiile de formă N fiind aceleași din (3.161).

Vectorul funcției necunoscute se scrie matriceal:

$$\{\Phi\}=\begin{Bmatrix} u \\ v \end{Bmatrix}=\begin{bmatrix} N_i & 0 & N_j & 0 & N_k & 0 \\ 0 & N_i & 0 & N_j & 0 & N_k \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix}=[N]\{\Phi\}_e \tag{3.164}$$

în care $[N]$ este matricea funcțiilor de formă.

Scanned with OKEN Scanner

Elementul finit triunghiular cu trei noduri fiind un element izoparametric, variația coordonatelor unui punct de pe elementul finit se exprimă în raport cu coordonatele nodale prin intermediul aceleiași matrice a funcțiilor de formă:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = [N] \begin{Bmatrix} x_i \\ y_i \\ x_j \\ y_j \\ x_k \\ y_k \end{Bmatrix} \tag{3.165}$$

Se prezintă în continuare funcții de formă pentru cele mai folosite tipuri de elemente finite liniare, pătratice și cubice.

3.8.3.2. Elemente finite unidimensionale

Exprimarea funcțiilor de formă pentru acest tip de elemente finite se face în sistemul de coordonate ξ — naturale (fig. 3.13):

— elementul liniar:

$$N_1 = \frac{1}{2}(1-\xi); \quad N_2 = \frac{1}{2}(1+\xi) \tag{3.166}$$

— elementul pătratic:

$$N_1 = -\frac{\xi}{2}(1-\xi); \quad N_2 = (1+\xi)(1-\xi); \quad N_3 = \frac{\xi}{2}(1+\xi) \tag{3.167}$$

— elementul cubic:

$$N_1 = -\frac{9}{16}(\xi-1)\left(\xi+\frac{1}{3}\right)\left(\xi-\frac{1}{3}\right); \quad N_2 = \frac{27}{16}(\xi+1)(\xi-1)\left(\xi-\frac{1}{3}\right);$$

$$N_3 = -\frac{27}{16}(\xi+1)(\xi-1)\left(\xi+\frac{1}{3}\right); \quad N_4 = \frac{9}{16}\left(\xi+\frac{1}{3}\right)\left(\xi-\frac{1}{3}\right)(\xi+1) \tag{3.168}$$

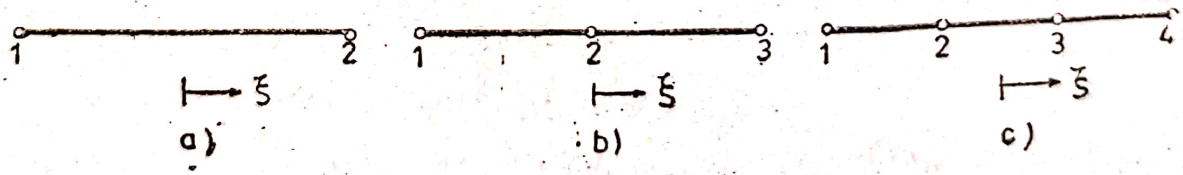


Fig. 3.13. Elemente finite unidimensionale:
a) liniar; b) pătratic; c) cubic

3.8.3.3. Elemente finite triunghiulare

Exprimarea funcțiilor de formă pentru acest tip de elemente finite se face în sistemul de coordonate L — naturale (fig. 3.14):

— elementul liniar:

$$N_1 = L_1; \quad N_2 = L_2; \quad N_3 = L_3 \tag{3.169}$$

Scanned with OKEN Scanner

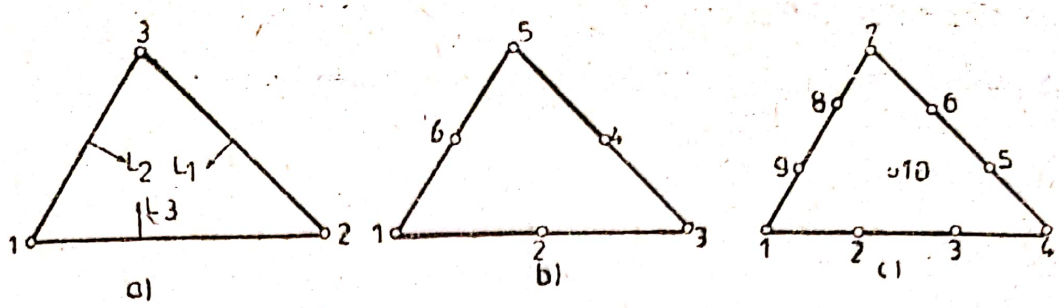


Fig. 3.14. Elemente finite triunghiulare:
a) liniar; b) pătratic; c) cubic

— elementul pătratic:

$$N_1=L_1(2L_1-1);\quad N_2=4L_1L_2;\quad N_3=L_2(2L_2-1);$$
$$N_4=4L_2L_3;\quad N_5=L_3(2L_3-1);\quad N_6=4L_1L_3 \tag{3.170}$$

— elementul cubic:

$$N_1=\frac{1}{2}L_1(3L_1-1)(3L_1-2);\quad N_2=\frac{9}{2}L_1L_2(3L_1-1);$$
$$N_3=\frac{9}{2}L_1L_2(3L_2-1);\quad N_4=\frac{1}{2}L_2(3L_2-1)(3L_2-2);$$
$$N_5=\frac{9}{2}L_2L_3(3L_2-1);\quad N_6=\frac{9}{2}L_2L_3(3L_3-1); \tag{3.171}$$
$$N_7=\frac{1}{2}L_3(3L_3-1)(3L_3-2);\quad N_8=\frac{9}{2}L_3L_1(3L_2-1);$$
$$N_9=\frac{9}{2}L_1L_3(3L_1-1);\quad N_{10}=27L_1L_2L_3$$

3.8.3.4. Elemente finite rectangulare serendipiene

Caracteristica de bază a acestor elemente este că nu prezintă noduri interne. Exprimarea funcțiilor de formă pentru acest tip de elemente finite se face în sistemul de coordonate ξ — naturale (fig. 3. 15):

— elementul liniar:

$$N_i=\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta);\qquad \xi_i,\ \eta_i=\pm 1 \tag{3.172}$$

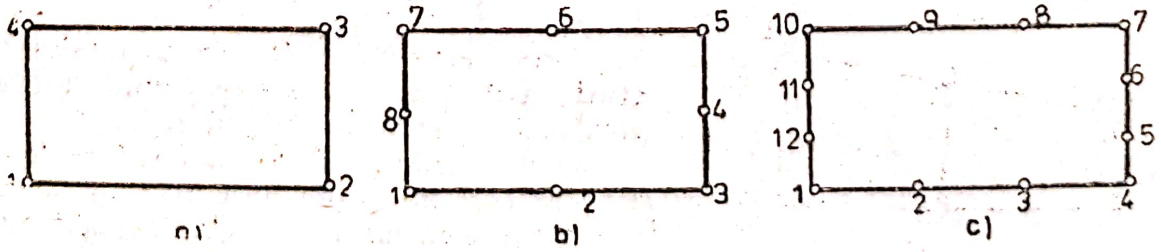


Fig. 3.15. Elemente finite rectangulare serendipiene:
a) liniar; b) pătratic; c) cubic

— elementul pătratic:

$$N_i=\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)(\xi_i\xi+\eta_i\eta-1),\ \text{pentru noduri la } \xi_i,\ \eta_i=\pm 1;$$

$$N_i=\frac{1}{2}(1-\xi^2)(1+\eta_i\eta),\ \text{pentru noduri la } \xi_i=0,\ \eta_i=\pm 1; \qquad (3.173)$$

$$N_i=\frac{1}{2}(1+\xi_i\xi)(1-\eta^2),\ \text{pentru noduri la } \xi_i=\pm 1,\ \eta_i=0$$

— elementul cubic:

$$N_i=\frac{1}{32}(1+\xi_i\xi)(1+\eta_i\eta)[9(\xi^2+\eta^2)-10],\ \text{pentru } \xi_i=\pm 1,\quad \eta_i=\pm 1 \qquad (3.174)$$

$$N_i=\frac{9}{32}(1+\xi_i\xi)(1-\eta^2)(1+9\eta_i\eta),\ \text{pentru } \xi_i=\pm 1,\ \eta_i=\pm 1/3.$$

3.8.3.5. Elemente finite patrulatere cu opt noduri

Elementul finit patrulater generalizează elementele finite bidimensionale (triughiular, rectangular), putînd fi aplicat cu uşurinţă oricărei geometrii a domeniului de analiză

Acest tip de element are opt noduri din care patru în colţurile elementului şi patru în mijloacele laturilor (fig. 3.16). Elementul este raportat la un sistem de coordonate ξ — naturale (cu originea în centrul de greutate al elementului), astfel încît laturile sale sînt reprezentate de ecuaţiile: $\xi=-1$, $\xi=1$, $\eta=-1$, $\eta=1$.

Funcţia de aproximare trebuie să fie un polinom care să aibă opt termeni, deoarece elementul are opt noduri şi să conţină toţi termenii de gradul întîi şi doi, întrucît pe o latură a elementului finit sînt trei noduri, iar ultimii doi termeni să fie de gradul trei, astfel încît elementul finit este cubic.

Funcţiile de formă pentru acest tip de element finit au expresiile:

$$N_1=-(1-\xi)(1-\eta)(1+\xi-\eta)/4; \qquad N_2=-(1+\xi)(1-\eta)(1-\xi+\eta)/4$$
$$N_3=-(1+\xi)(1+\eta)(1-\xi-\eta)/4; \qquad N_4=-(1-\xi)(1+\eta)(1+\xi-\eta)/4$$
$$N_5=(1-\xi^2)(1-\eta)/2; \qquad N_6=(1-\xi^2)(1+\eta)/2 \qquad (3.175)$$
$$N_7=(1+\xi)(1+\eta^2)/2; \qquad N_8=(1-\xi)(1-\eta^2)/2$$

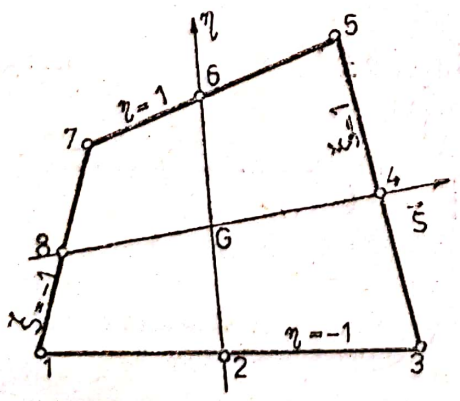


Fig. 3.16. Element finit patrulater cu opt noduri

Geometria bidimensională a elementului finit se exprimă prin relaţiile:

$$x=\sum_{i=1}^{8}N_i x_i; \qquad y=\sum_{i=1}^{8}N_i y_i \qquad (3.176)$$

Acest element finit are avantajul că laturile pot fi curbe, ceea ce permite modelarea mai fidelă a domeniului analizat.

Pentru îmbunătăţirea preciziei se foloseşte deseori un al nouălea nod amplasat în centrul elementului, a cărui funcţie de formă este:

$$N_9=(1-\xi^2)(1-\eta^2) \qquad (3.177)$$

Scanned with OKEN Scanner

3.8.3.6. Elemente finite curbilinii

Unul dintre avantajele primordiale ale metodei cu elemente finite este acela de a putea discretiza forme geometrice complexe, incluzînd cele cu laturi şi feţe curbilinii. Pentru discretizarea elementelor curbilinii au fost create o serie de elemente finite cu laturi curbilinii. Aceste elemente pot fi obţinute din elemente prezentate anterior, prin transformări geometrice convenabile. În figura 3.17 se arată cum se pot obţine elemente triunghiulare şi patrulatere cu laturi curbilinii. Transformarea coordonatelor se face pe baza relaţiilor:

$$x=\sum_{i=1}^{n}N_i x_i; \quad y=\sum_{i=1}^{n}N_i y_i; \quad z=\sum_{i=1}^{n}N_i z_i \tag{3.178}$$

în care: x_i, y_i, z_i sînt coordonate carteziene nodale; N_i — funcţii de formă corespunzătoare, exprimate în coordonate ξ — naturale; n — numărul de noduri ale elementului.

Dacă aceste transformări au acelaşi ordin de mărime ca şi al reprezentării variabilelor de cîmp, elementele finite se numesc *izoparametrice*. Deci, elementele izoparametrice folosesc aceleaşi funcţii de formă pentru coordonate şi pentru reprezentarea variabilelor de cîmp.

Dacă ordinul transformărilor geometrice este superior celui al reprezentării variabilelor de cîmp, elementele sînt numite *supraparametrice*, iar dacă situaţia ordinelor celor două reprezentări este inversă, elementele se numesc *subparametrice*.

Evaluarea derivatelor $\partial N_i/\partial x$, $\partial N_i/\partial y$ se face cu ajutorul transformării:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} \tag{3.179}$$

unde $[J]$ este Jacobianul transformării.

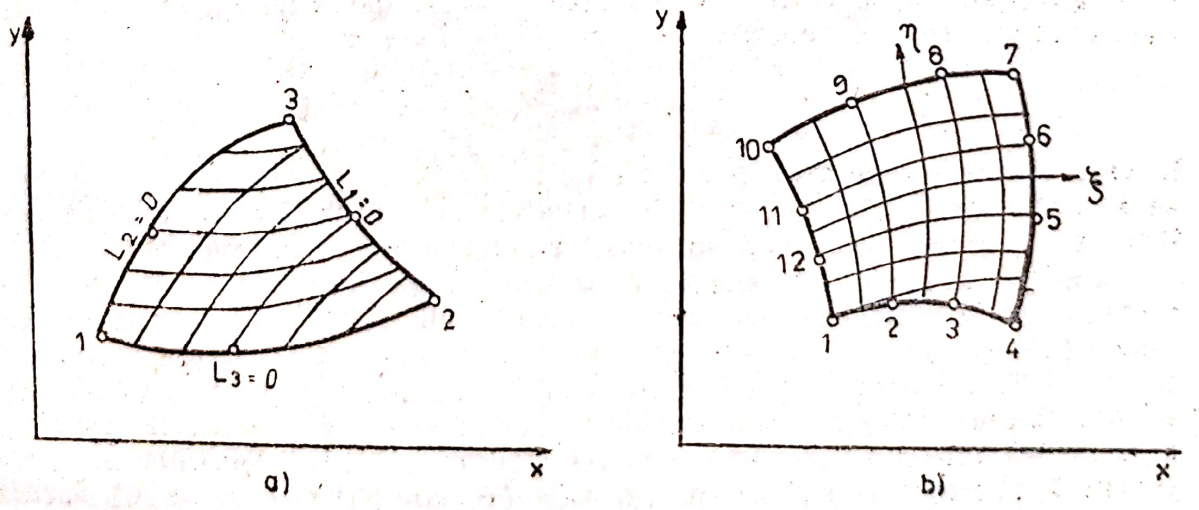


Fig. 3.17. Element finit cu laturi curbilinii

3.8.4. MODELUL ANALITIC AL CONDUCȚIEI TERMICE

Temperatura dintr-un corp este o funcție de coordonatele spațiului și de timp.

Totalitatea punctelor din interiorul unui corp care au aceeași temperatură se află pe o suprafață izotermă, care în cazul cîmpurilor bidimensionale devine o curbă izotermă.

Vectorul dirijat după normala de suprafața izotermă, în sensul creșterii temperaturii și egal cu derivata temperaturii în raport cu această direcție, se numește *gradient de temperatură*:

$$\text{grad } t = \bar{n}_0 \frac{\partial t}{\partial n} \tag{3.180}$$

în care $\bar{n}_0$ este vectorul unitar al normalei la suprafața izotermă.

Cantitatea de căldură care trece în unitatea de timp printr-o suprafață izotermă S se numește *flux de căldură* și are expresia:

$$Q = \int_S q \, ds \tag{3.181}$$

unde q este fluxul termic unitar, care conform legii experimentale a lui Fourier are expresia:

$$q = -\lambda \frac{\partial t}{\partial n} = -\lambda \text{ grad } t \tag{3.182}$$

în care λ este conductivitatea termică a materialului.

Pentru majoritatea materialelor variația coeficientului λ cu temperatura poate fi aproximată prin relația liniară:

$$\lambda = \lambda_0 [1 + b(t - t_0)] \tag{3.183}$$

în care: λ_0 este conductivitatea termică la temperatura t_0; b — constantă ce depinde de material.

Se consideră cazul general de transfer de căldură prin conducție într-un mediu neomogen și anizotrop, ceea ce înseamnă că mediul respectiv poate fi format din regiuni cu materiale diferite, iar materialele pot fi considerate ca avînd proprietăți ce se propagă diferit în direcții diferite. De asemenea se presupune că mediul este liniar, deci coeficientul de conductivitate termică este constant.

Variația temperaturii în spațiu și timp în interiorul unui astfel de mediu neomogen, anizotrop și liniar se exprimă prin *ecuația diferențială a conducției* în forma cea mai generală, sau *ecuația lui Fourier*:

$$\rho c \frac{\partial t}{\partial \tau} = \frac{\partial}{\partial x}\left(\lambda_x \frac{\partial t}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda_y \frac{\partial t}{\partial y}\right) + \frac{\partial}{\partial z}\left(\lambda_z \frac{\partial t}{\partial z}\right) + Q_0 \tag{3.184}$$

în care: t este temperatura; τ — timpul; ρ, c — densitatea și căldura specifică a materialului; $\lambda_x, \lambda_y, \lambda_z$ — componentele conductivității termice pe cele trei axe x, y și z; Q_0 — debitul surselor interioare de căldură.

Analiza numerică cu elemente finite a cîmpurilor termice constituie o metodologie specifică de lucru pentru problemele de stare ale sistemelor termice. Formularea lor corectă și completă impune cunoașterea geometriei mediului de analizat, a proprietăților fizice ale materialelor constituente și a interacțiunii dintre sistemul termic și mediul ambiant. Această interacțiune se specifică din punct de vedere matematic sub forma condițiilor la limită (spațială) și respectiv a condițiilor inițiale (de timp) pentru cazul regimurilor tranzitorii.

Scanned with OKEN Scanner

Condițiile inițiale stabilesc distribuția de temperatură la începutul procesului:

$$t=f(x,\ y,\ z,\tau=0)=f(x,\ y,\ z) \qquad (3.185)$$

Condițiile la limită spațiale stabilesc parametrii schimbului de căldură pe frontierele mediului studiat, fiind de următoarele tipuri:

— *condiții Dirichlet* (de tipul întîi). În acest caz se specifică temperatura pe segmentul de suprafață S_t al mediului analizat (fig. 3.18) ca o funcție spațială constantă sau variabilă în timp:

$$t=g(x,\ y,\ z,\ \tau) \qquad (3.186)$$

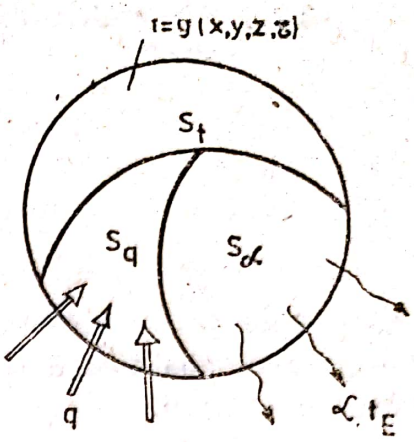


Fig. 3.18. Condițiile la limită spațiale

— *condiții Neumann* (de tipul al doilea). În acest caz se specifică valoarea fluxului termic unitar de suprafață pe segmentul de front eră S_q.

$$q=\lambda_x\frac{\partial t}{\partial x}n_x+\lambda_y\frac{\partial t}{\partial y}n_y+\lambda_z\frac{\partial t}{\partial z}n_z, \qquad (3.187)$$

în care n_x, n_y, n_z sînt cosinusurile directoare ale normalei la suprafața S_q

— *condiții Cauchy* (de tipul al treilea). În acest caz se specifică temperatura mediului exterior și valoarea coeficientului de transfer convectiv între segmentul de frontieră S_α și fluidul înconjurător:

$$\alpha(t-t_E)=\lambda_x\frac{\partial t}{\partial x}n_x+\lambda_y\frac{\partial t}{\partial y}n_y+\lambda_z\frac{\partial t}{\partial z}n_z \qquad (3.188)$$

în care: α este coeficientul de transfer termic convectiv de la segmentul de frontieră S_α la mediul înconjurător sau invers; t_E — temperatura mediului exterior.

Modelul analitic (3.184)...(3.188) trebuie completat cu o ecuație de material, care descrie variația proprietăților fizice în funcție de temperatură. Pentru corpurile cu proprietăți fizice liniare, aceste ecuații sînt de forma $\lambda=$const., fapt pentru care ele nu se mai scriu în cadrul modelului.

Integrarea ecuației diferențiale a conducției termice (3.184) în condițiile la limită corespunzătoare problemei de rezolvat permite determinarea temperaturii în orice punct al mediului și în orice moment. Integrarea analitică este posibilă doar în cazul problemelor simple (conducția prin pereți plani, cilindrici, sferici, simpli sau compuși), fiind foarte dificilă și chiar imposibilă în cazul domeniilor cu forme geometrice complicate, cînd se apelează la integrarea numerică.

Pentru realizarea modelului numeric cu elemente finite, modelul analitic (3.184)...(3.188) trebuie transformat într-un model integral adecvat domeniului de analiză considerat. Această transformare se va face folosind *metoda variațională*, care prezintă avantajul unei interpretări fizice a fenomenului modelat.

Scanned with OKEN Scanner

3.8.5. FORMULAREA VARIAȚIONALĂ A METODEI ELEMENTELOR FINITE

Funcția de temperatură $t(x, y, z, \tau)$ ce satisface ecuația fundamentală a transferului de căldură (3.184) și condițiile (3.185)...(3.188) va satisface și condiția de staționaritate în domeniul de analiză V:

$$\delta\Gamma = 0 \tag{3.189}$$

care reprezintă o formulare echivalentă din punct de vedere matematic cu formularea analitică dată de (3.184)...(3.188), unde Γ este funcționala conducției căldurii.

Deci, integrarea ecuației diferențiale a conducției căldurii (3.184), în condițiile la limită (3.185)...(3.188), este echivalentă cu minimizarea funcționalei:

$$\Gamma = \int_V \frac{1}{2}\left[\lambda_x\left(\frac{\partial t}{\partial x}\right)^2 + \lambda_y\left(\frac{\partial t}{\partial y}\right)^2 + \lambda_z\left(\frac{\partial t}{\partial z}\right)^2\right] dV + \int_V \left(\rho c\frac{\partial t}{\partial \tau} - Q_0\right) t\, dV - \int_{S_q} q t\, dS + \int_{S_\alpha} \alpha t\left(\frac{1}{2}t - t_E\right) dS \tag{3.190}$$

Convenția pentru semnele lui Q_0, q și α este următoarea: Q_0 este pozitiv dacă sursele interioare debitează căldură și negativ dacă absorb căldură; q este pozitiv cînd pătrunde în corp dinspre mediul exterior și negativ cînd iese din corp spre mediul exterior; α este pozitiv pe suprafețele care cedează căldură mediului exterior și negativ pe cele ce absorb căldură de la mediul exterior.

Prin particularizarea funcționalei (3.190) se obțin cazurile frecvent întîlnite în practica modelării cu elemente finite a cîmpurilor termice:

— regim termic staționar fără surse interne de căldură:

$$\Gamma = \int_V \frac{1}{2}\left[\lambda_x\left(\frac{\partial t}{\partial x}\right)^2 + \lambda_y\left(\frac{\partial t}{\partial y}\right)^2 + \lambda_z\left(\frac{\partial t}{\partial z}\right)^2\right] dV - \int_{S_q} q\, t\, dS + \int_{S_\alpha} \alpha t\left(\frac{1}{2}t - t_E\right) dS \tag{3.191}$$

— regim termic staționar omogen și izotrop, fără surse interne de căldură:

$$\Gamma = \int_V \frac{\lambda}{2}\left[\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2\right] dV - \int_{S_q} q\, t\, dS + \int_{S_\alpha} \alpha t\left(\frac{1}{2}t - t_E\right) dS \tag{3.192}$$

— regim termic staționar omogen și izotrop, fără surse interne de căldură și condiții la limită de tip Dirichlet:

$$\Gamma = \int_V \frac{1}{2}\left[\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2\right] dV \tag{3.193}$$

Minimizarea funcționalei se face pe elemente finite, efectul de asamblare obținîndu-se prin cumularea contribuției fiecărui element finit în parte. În fiecare nod există un singur parametru care este temperatura, mărime scalară.

Scanned with OKEN Scanner

3.8.6. MODELUL NUMERIC CU ELEMENTE FINITE AL CONDUCȚIEI TERMICE BIDIMENSIONALE ÎN REGIM STAȚIONAR

În transferul staționar de căldură, temperatura este constantă în timp și, în consecință, în (3.190), $\partial t/\partial \tau = 0$. Pentru probleme bidimensionale temperatura nu variază în direcția axei z, încît $\partial t/\partial z = 0$. Ca probleme bidimensionale se studiază problemele plane.

3.8.6.1. Forma generală a ecuației elementelor finite

În acest caz funcționala (3.190) primește forma:

$$\Gamma = \int_V \left\{ \frac{1}{2}\left[\lambda_x \left(\frac{\partial t}{\partial x}\right)^2 + \lambda_y \left(\frac{\partial t}{\partial y}\right)^2 \right] - Q_0 t \right\} \mathrm{d}V - \int_{S_q} q\, t\, \mathrm{d}S + \int_{S_\alpha} \alpha t \left(\frac{1}{2} t - t_E \right) \mathrm{d}S \qquad (3.194)$$

Deoarece funcția convențională prin care se aproximează temperatura nu este continuă pe tot domeniul ci numai pe cuprinsul elementelor finite, integrala din (3.194) trebuie efectuată pe elemente finite. Funcționala Γ pentru toată structura se scrie ca o sumă de funcționale pentru m elemente finite în care a fost divizat domeniul:

$$\Gamma = \sum_{e=1}^{m} \Gamma^e \qquad (3.195)$$

$$\Gamma = \sum_{e=1}^{m} \left\{ \int_{V_e} \frac{1}{2}\left[\lambda_x \left(\frac{\partial t^e}{\partial x}\right)^2 + \lambda_y \left(\frac{\partial t^e}{\partial y}\right)^2 \right] \mathrm{d}V - \int_{V_e} Q_0 t^e \, \mathrm{d}V - \int_{S_{qe}} q t^e \mathrm{d}S + \right.$$

$$\left. + \int_{S_{\alpha e}} \alpha t^e \left(\frac{1}{2} t^e - t_E \right) \mathrm{d}S \right\} \qquad (3.196)$$

În relația (3.196) exponentul „e" precizează referirea la un element finit. Pentru simplificarea scrierii el a fost folosit numai în cazul temperaturii, deși celelalte mărimi care intervin în relație se referă tot la un element finit.

Pe cuprinsul unui element finit temperatura t^e în orice punct se exprimă în raport cu temperaturile nodale printr-o relație de forma (3.160):

$$t^e = N_1 t_1 + N_2 t_2 + \ldots + N_n t_n = [N]\,\{t\}_e \qquad (3.197)$$

în care: n este numărul nodurilor elementului finit; $[N]$ — matricea funcțiilor de formă pe elementul finit; $\{t\}_e$ — vectorul valorilor nodale ale temperaturii.

În funcționala (3.196) intervin și derivatele parțiale ale temperaturii, se derivează (3.197) în raport cu x și y și se obține relația:

$$B = \begin{Bmatrix} \dfrac{\partial t^e}{\partial x} \\ \dfrac{\partial t^e}{\partial y} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial x} \ldots \dfrac{\partial N_n}{\partial x} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_2}{\partial y} \ldots \dfrac{\partial N_n}{\partial y} \end{bmatrix} \begin{Bmatrix} t_1 \\ t_2 \\ t_n \end{Bmatrix} = [J]\,\{t\}_e \qquad (3.198)$$

Dacă se notează matricea conductivităților termice:

$$[D] = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_y \end{bmatrix} \qquad (3.199)$$

Scanned with OKEN Scanner

funcționala (3.196) se scrie, pentru un element finit sub forma:

$$\Gamma^e=\int_{V_e}\frac{1}{2}([J]\{t\}_e)^T[D][J]\{t\}_e\,dV-\int_{V_e}Q_0[N]\{t\}_e\,dV-\int_{S_{qe}}q[N]\{t\}_e dS+$$

$$+\int_{S_{\alpha e}}\frac{\alpha}{2}([N]\{t\}_e)^2\,dS-\int_{S_{\alpha e}}\alpha t_E[N]\{t\}_e dS \qquad (3.200)$$

Deoarece:

$$([J]\{t\}_e)^T=\{t\}_e^T[J]^T$$

$$([N]\{t\}_e)^2=([N]\{t\}_e)^T([N]\{t\}_e)=\{t\}_e[N]^T[N]\{t\}_e, \qquad (3.201)$$

ecuația (3.200) se scrie:

$$\Gamma^e=\int_{V_e}\frac{1}{2}\{t\}_e^T[J]^T[D][J]\{t\}_e\,dV-\int_{V_e}Q_0[N]\{t\}_e dV-\int_{S_{qe}}q[N]\{t\}_e\,dS+$$

$$+\int_{S_{\alpha e}}\frac{\alpha}{2}\{t\}_e^T[N]^T[N]\{t\}_e\,dS-\int_{S_{\alpha e}}\alpha t_E[N]\{t\}_e dS \qquad (3.202)$$

Aplicînd regulile de derivare a expresiilor matriceale, se obține:

$$\frac{\partial\Gamma^e}{\partial\{t\}_e}=\left(\int_{V_e}[J]^T[D][J]\,dV+\int_{S_{\alpha e}}\alpha[N]^T[N]\,dS\right)\{t\}_e-\int_{V_e}Q_0[N]^T dV-$$

$$-\int_{S_{qe}}q[N]^T\,dS-\int_{S_{\alpha e}}\alpha t_E[N]^T dS \qquad (3.203)$$

Ținînd seama că $dV=h\,dA$ și $dS=h\,dL$, unde h este grosimea elementului finit, dA — elementul diferențial de arie, dL — elementul diferențial de lungime, rezultă:

$$\frac{\partial\Gamma^e}{\partial\{t\}_e}=h\left(\int_{A_e}[J]^T[D][J]\,dA+\int_{L_e}\alpha[N]^T[N]\,dL\right)\{t\}_e-h\int_{A_e}Q_0[N]^T\,dA-$$

$$-h\int_{L_e}q[N]^T\,dL-h\int_{L_e}\alpha t_E[N[^T dL \qquad (3.204)$$

Grosimea h a elementului finit se consideră constantă, egală cu unitatea.

Concentrat, relația (3.204) se scrie:

$$\frac{\partial\Gamma^e}{\partial\{t\}_e}=[k]\{t\}_e-\{f\} \qquad (3.205)$$

unde:

$$[k]=h\int_{A_e}[J]^T[D][J]dA+h\int_{L_{\alpha e}}\alpha[N]^T[N]\,dL \qquad (3.206)$$

$$\{f\}=h\int_{A_e}Q_0[N]^T dA+h\int_{L_{qe}}q[N]^T dL+h\int_{L_{\alpha e}}\alpha t_E[N]^T dL \qquad (3.207)$$

Scanned with OKEN Scanner

în care: $[k]$ — este matricea de conductivitate, prima integrală a acesteia corespunzînd conducției în corp, iar în a doua integrală corespunzînd convecției pe latura L_{α_e} a suprafeței S_{α_e}; $\{f\}$ — vectorul încărcării termice, în cadrul căruia integralele reprezintă contribuția respectiv a surselor interioare de căldură Q_0, pe elementul finit de volum V_e, a fluxului termic unitar q pe segmentul de frontieră S_{q_e}, și a convecției termice pe suprafața S_α.

Minimizarea funcționalei Γ se obține anulînd derivata în raport cu temperaturile din toate nodurile domeniului. Ținînd seama de (3.195) se obține:

$$\frac{\partial \Gamma}{\partial \{t\}} = \frac{\partial}{\partial \{t\}} \sum_{e=1}^{m} \Gamma^e = \sum_{e=1}^{m} \frac{\partial \Gamma}{\partial \{t\}_e} \tag{3.208}$$

Introducînd (3.205) în (3.208) se obține sistemul de ecuații al întregului domeniu:

$$[K]\{t\} = \{F\} \tag{3.209}$$

unde:

$$[K] = \sum_{1}^{m} [k]; \quad \{F\} = \sum_{1}^{m} \{f\} \tag{3.210}$$

în care: $[K]$ este matricea de conductivitate a domeniului; $\{F\}$ — vectorul încărcării termice pentru întreg domeniul; $\{t\}$ — vectorul temperaturilor necunoscute pentru tot domeniul.

Ecuația (3.209) este forma cu elemente finite a ecuației diferențiale a conducției căldurii, reprezentînd un sistem de atîtea ecuații cîte noduri cu temperatură necunoscută are domeniul analizat, prin a cărui rezolvare se determină aceste temperaturi.

În cele ce urmează se determină expresia matricei de conductivitate (3.206) și a vectorului încărcării termice pe elementul finit (3.207).

3.8.6.2. Expresia matricei de conductivitate

Folosind elemente finite triunghiulare cu trei noduri (liniare), temperatura t^e într-un punct al elementului finit este, conform (3.197):

$$t^e = N_i t_i + N_j t_j + N_k t_k = [N_i N_j N_k] \begin{Bmatrix} t_i \\ t_j \\ t_k \end{Bmatrix} = [N]\{t\}_e \tag{3.211}$$

în care: t_i, t_j, t_k sînt temperaturile din nodurile i, j, k ale elementului finit (fig. 3.12), iar $[N]$ — matricea funcțiilor de formă. Funcțiile de formă N_i N_j, N_k au expresiile (3.161).

Matricea de conductivitate a elementului finit (3.206) se scrie sub forma:

$$[k] = [k_1] + [k_2] \tag{3.212}$$

în care:

$$[k_1] = h \int_{A_e} [J]^T [D][J]\, dA; \quad [k_2] = h\alpha \int_{L_{\alpha_e}} [N]^T [N] dL \tag{3.213}$$

Scanned with OKEN Scanner

Matricea $[J]$ se obține din relația de definiție (3.198), care pentru acest tip de element finit se scrie:

$$\{B\}=\begin{Bmatrix}\frac{\partial t^e}{\partial x}\\ \frac{\partial t^e}{\partial y}\end{Bmatrix}=\begin{bmatrix}\frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \frac{\partial N_k}{\partial x}\\ \frac{\partial N_i}{\partial y} & \frac{\partial N_j}{\partial y} & \frac{\partial N_k}{\partial y}\end{bmatrix}\begin{Bmatrix}t_i\\ t_j\\ t_k\end{Bmatrix}=[J]\{t\}_e \tag{3.214}$$

Derivînd expresiile (3.161) ale funcțiilor de formă, se obține:

$$[J]=\begin{vmatrix}\frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \frac{\partial N_k}{\partial x}\\ \frac{\partial N_i}{\partial y} & \frac{\partial N_j}{\partial y} & \frac{\partial N_k}{\partial y}\end{vmatrix}=\frac{1}{2A_e}\begin{bmatrix}b_i & b_j & b_k\\ c_i & c_j & c_k\end{bmatrix} \tag{3.215}$$

în care b și c sînt coeficienți constanți dați de (3.159), iar A_e este aria elementului finit conform (3.158). Rezultă că matricea $[J]$ este constantă. Deoarece conductivitățile termice λ_x și λ_y sînt considerate constante pe elementul finit, matricea $[D]$ din relația (3.199) este și ea constantă, încît:

$$[k_1]=h\int_{A_e}[J]^T[D][J]\,dA=h[J]^T[D][J]A_e \tag{3.216}$$

Înlocuind pe $[J]$ din (3.215) și pe $[D]$ din (3.199) în (3.216 se obține:

$$[k_1]=\frac{h}{4A_e}\begin{bmatrix}\lambda_x b_i b_i+\lambda_y c_i c_i & \lambda_x b_i b_j+\lambda_y c_i c_j & \lambda_x b_i b_k+\lambda_y c_i c_k\\ \lambda_x b_j b_i+\lambda_y c_j c_i & \lambda_x b_j b_j+\lambda_y c_j c_j & \lambda_x b_j b_k+\lambda_y c_j c_k\\ \lambda_x b_k b_i+\lambda_y c_k c_i & \lambda_x b_k b_j+\lambda_y c_k c_j & \lambda_x b_k b_k+\lambda_y c_k c_k\end{bmatrix} \tag{3.217}$$

Matricea $[k_2]$ din (3.212) se dezvoltă sub forma:

$$[k_2]=h\alpha\int_{L_{\alpha e}}\begin{bmatrix}N_iN_i & N_iN_j & N_iN_k\\ N_jN_i & N_jN_j & N_jN_k\\ N_kN_i & N_kN_j & N_kN_k\end{bmatrix}dL \tag{3.218}$$

Introducînd coordonatele L — naturale descrise la punctul 3.8.1.2 și considerînd că schimbul de căldură prin convecție se face pe latura jk a elementului finit se obține:

$$[k_2]=h\alpha\int_{L_{\alpha e}}\begin{bmatrix}0 & 0 & 0\\ 0 & L_jL_j & L_jL_k\\ 0 & L_jL_k & L_kL_k\end{bmatrix}dL \tag{3.219}$$

Pentru calculul valorii integralei din (3.219) se folosește formula (3.141). Pentru produsele cu aceiași indici j sau k se obține:

$$\int_{L_{\alpha e}}L_jL_j\,dL=\int_{L_{\alpha e}}L_kL_k\,dL=\int_{L_{\alpha e}}L_j^2\,dL=\frac{2!0!}{(2+0+1)!}L_{\alpha e}=\frac{L_{\alpha e}}{3} \tag{3.220}$$

iar pentru produsele cu indici j și k, rezultă:

$$\int_{L_{\alpha e}}L_jL_k\,dL=\int_{L_{\alpha e}}L_kL_j\,dL=\frac{1!1!}{(1+1+1)!}L_{\alpha e}=\frac{L_{\alpha e}}{6} \tag{3.221}$$

Scanned with OKEN Scanner

Înlocuind în (3.219) se obține:

$$[k_2]=\frac{h\alpha L_{\alpha e}}{6}\begin{bmatrix}0&0&0\\0&2&1\\0&1&2\end{bmatrix} \tag{3.222}$$

Dacă latura elementului finit cu schimb de căldură prin convecție este ij respectiv ki, se obține:

$$[k_2]=\frac{h\alpha L_{\alpha e}}{6}\begin{bmatrix}2&1&0\\1&2&0\\0&0&0\end{bmatrix};\quad [k_2]=\frac{h\alpha L_{\alpha e}}{6}\begin{bmatrix}2&0&1\\0&0&0\\1&0&2\end{bmatrix} \tag{3.223}$$

Matricea $[k_2]$ se constituie numai dacă elementul respectiv are schimb de căldură prin convecție pe una din laturile sale.

3.8.6.3. Expresia vectorului încărcării termice

Acest vector se compune, conform (3.207), din trei termeni care se calculează cu ajutorul coordonatelor L — naturale. Primul termen corespunde debitului surselor interioare de căldură pe elementul finit. Considerînd debitul Q_0 constant pe elementul finit și folosind formula de integrare (3.144), se obține:

$$\{f_Q\}=h\int_{A_e}Q_0[N]^T\,dA=hQ_0\int_{A_e}\begin{Bmatrix}N_i\\N_j\\N_k\end{Bmatrix}dA=hQ_0\int_{A_e}\begin{Bmatrix}L_i\\L_j\\L_k\end{Bmatrix}dA=\frac{hQ_0A_e}{3}\begin{Bmatrix}1\\1\\1\end{Bmatrix} \tag{3.224}$$

Al doilea termen din (3.207) corespunde schimbului de căldură pe contur prin flux termic impus. Presupunînd că fluxul de căldură pătrunde prin latura $L_{ki}=L_{qe}$ a elementului finit și folosind formula de integrare (3.141), se obține:

$$\{f_q\}=h\int_{L_{qe}}q[N]^T\,dL=hq\int_{L_{qe}}\begin{Bmatrix}N_i\\0\\N_k\end{Bmatrix}dL=hq\int_{L_{qe}}\begin{Bmatrix}L_i\\0\\L_k\end{Bmatrix}dL=\frac{hqL_{qe}}{2}\begin{Bmatrix}1\\0\\1\end{Bmatrix} \tag{3.225}$$

Al treilea termen din (3.207) corespunde schimbului de căldură prin convecție, care se presupune că se produce pe latura jk de lungime $L_{jk}=L_{\alpha e}$ a elementului finit. Folosind formula de integrare (3.141) se obține:

$$\{f_\alpha\}=h\int_{L_{\alpha e}}\alpha t_E[N]^T\,dL=h\alpha t_E\int_{L_{\alpha e}}\begin{Bmatrix}0\\N_j\\N_k\end{Bmatrix}dL=h\alpha t_E\int_{L_{\alpha e}}\begin{Bmatrix}0\\L_j\\L_k\end{Bmatrix}dL+\frac{h\alpha t_EL_{\alpha e}}{2}\begin{Bmatrix}0\\1\\1\end{Bmatrix} \tag{3.226}$$

De menționat că elementul nul din vectorii din expresiile (3.225) și (3.226) poate ocupa orice poziție în vector, după cum nodul opus laturii cu schimb de căldură prin flux termic sau prin convecție este nodul i, j sau k.

Cu ajutorul expresiilor deduse anterior se poate constitui sistemul de ecuații al elementului finit, care apoi se asamblează în sistemul de ecuații al domeniului analizat (3.207), ce se poate rezolva folosind metoda Cholesky.

Scanned with OKEN Scanner

3.8.7. MODELUL NUMERIC CU ELEMENTE FINITE AL CONDUCȚIEI TERMICE BIDIMENSIONALE ÎN REGIM TRANZITORIU

Comparînd funcționala (3.190) valabilă pentru cazul general de transfer termic cu funcționala (3.194) din cazul regimului staționar, se observă că, menținînd analiza în cadrul problemelor plane, $\partial t/\partial z=0$, deosebirea dintre cele două funcționale constă în faptul că în locul termenului $-\int_V Q_0 t\,dV$ de la regimul staționar (3.194) apare termenul:

$$\int_V \left(\rho c\frac{\partial t}{\partial \tau}-Q_0\right) t\,dV=\int_V \rho c\frac{\partial t}{\partial \tau}t\,dV-\int_V Q_0 t\,dV, \tag{3.227}$$

la regim tranzitoriu.

Deci pentru trecerea de la regimul staționar la cel tranzitoriu trebuie adăugată contribuția rezultată prin minimizarea termenului:

$$\gamma=\int_V (\rho c)_e\frac{\partial t}{\partial \tau}t\,dV \tag{3.228}$$

Deoarece minimizarea se face tot pe elemente finite, pentru un element finit vom avea:

$$\gamma^e=\int_{V_e}(\rho c)\frac{\partial t^e}{e\partial \tau}t_e\,dV \tag{3.229}$$

Exprimînd temperatura t^e prin intermediul funcțiilor de formă conform (3.197), în care $[N]$ este independent de timpul τ, se obține:

$$\frac{\partial t^e}{\partial \tau}=[N]\frac{\partial\{t\}_e}{\partial \tau} \tag{3.230}$$

Introducînd (3.197) și (3.230) în (3.229), rezultă:

$$\gamma^e=\int_V (\rho c)_e[N]^T\{t\}_e[N]\,dV\frac{\partial\{t\}_e}{\partial \tau} \tag{3.231}$$

Pentru a minimiza pe γ este necesar ca $\partial\gamma^e/\partial\{t\}_e=0$, ceea ce conduce la:

$$\left(\int_{V_e}(\rho c)_e[N]^T[N]\,dV\right)\frac{\partial\{t\}_e}{\partial \tau}=0 \tag{3.232}$$

Matricea

$$[C]=\int_V(\rho c)_e[N]^T[N]\,dV \tag{3.233}$$

se numește matricea capacităților calorice pentru un element finit.

Deci pentru trecerea de la regimul staționar la cel tranzitoriu, ecuația elementului finit (3.205) trebuie completată cu termenul $[C](\partial\{t\}_e/\partial\tau)$, devenind:

$$[C]\frac{\partial\{t\}_e}{\partial \tau}+[k]\{t\}_e-\{f\}=0 \tag{3.234}$$

În locul unui sistem de ecuații liniare se obține un sistem de ecuații diferențiale liniare de ordinul întîi în raport cu timpul.

Scanned with OKEN Scanner

Admițînd că pe parcursul unui element finit mărimile ρ și c sînt constante, efectuarea integralei (3.233) cu folosirea formulei (3.144) conduce pentru elementul finit triunghiular plan la expresia:

$$[C]=h\frac{(\rho c)_e A_e}{12}\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix} \tag{3.235}$$

Pentru integrarea în raport cu timpul a sistemului de ecuații diferențiale (3.234) și determinarea cîmpului de temperatură tranzitoriu, se folosește *metoda reziduurilor* (Galerkin).

Alegînd la nivelul unui element coeficienții de pondere $W_k=N_{\tau k}$, condiția generală specifică metodei reziduurilor este:

$$\int_D N_{\tau k}R(t)\,\mathrm{d}D \tag{3.236}$$

în care:

$$R(t)=[C]\frac{\partial\{t\}_e}{\partial\tau}+[k]\{t\}_e-\{f\} \tag{3.237}$$

Domeniul de integrare D este pasul de timp $\Delta\tau$, încît condiția (3.236) se scrie:

$$\int_0^{\Delta\tau}[N_\tau]^T R\{t\}\,\mathrm{d}t=\int_0^{\Delta\tau}[N_\tau]^T\left([C]\frac{\partial\{t\}_e}{\partial\tau}+[k]\{t\}_e-\{f\}\right)\mathrm{d}t=0 \tag{3.238}$$

Funcțiile de forma $[N_\tau]$ descriu variația temperaturii în intervalul de timp $\Delta\tau$, în raport cu temperaturile la începutul și respectiv la sfîrșitul intervalului de timp, la fel cum funcțiile de formă $[N]$ descriu variația temperaturii pe suprafața elementului finit în raport cu temperaturile nodale.

Dacă se notează $\{t\}_{ei}$ vectorul temperaturilor din nodurile elementului finit la începutul intervalului de timp și $\{t\}_{ej}$ — la sfîrșitul intervalului de timp, temperatura în nodurile elementului la un moment cuprins între începutul τ_i și sfîrșitul τ_j al intervalului de timp, este:

$$\{t\}_e=[N_\tau]\begin{Bmatrix}\{t\}_{ei}\\\{t\}_{ej}\end{Bmatrix}=[N_{\tau i}N_{\tau j}]\begin{Bmatrix}\{t\}_{ei}\\\{t\}_{ej}\end{Bmatrix}=N_{\tau i}\{t\}_{ei}+N_{\tau j}\{t\}_{ej} \tag{3.239}$$

Admițînd o variație liniară a temperaturii în raport cu timpul, funcțiile de formă vor fi funcții liniare de timp:

$$N_{\tau i}=1-\frac{1}{\Delta\tau};\qquad N_{\tau i}=\frac{1}{\Delta\tau} \tag{3.240}$$

Derivata temperaturii în raport cu timpul este:

$$\frac{\partial\{t\}}{\partial\tau}=\begin{bmatrix}-\frac{1}{\Delta t}&\frac{1}{\Delta t}\end{bmatrix}\begin{Bmatrix}\{t\}_{ei}\\\{t\}_{ej}\end{Bmatrix} \tag{3.241}$$

Introducînd (3.239) și (3.241) în (3.238) și efectuînd integrarea se obține sistemul de ecuații matriceale:

$$\begin{bmatrix}\left(-\frac{1}{2}[C]+\frac{\Delta\tau}{3}[k]\right)&\left(\frac{1}{2}[C]+\frac{\Delta\tau}{6}[k]\right)\\\left(-\frac{1}{2}[C]+\frac{\Delta\tau}{6}[k]\right)&\left(\frac{1}{2}[C]+\frac{\Delta\tau}{3}[k]\right)\end{bmatrix}\begin{Bmatrix}\{t\}_{ei}\\\{t\}_{ej}\end{Bmatrix}-\frac{\Delta\tau}{2}\begin{Bmatrix}\{f\}_i\\\{f\}_j\end{Bmatrix}=0 \tag{3.242}$$

Scanned with OKEN Scanner

Pentru rezolvarea pas cu pas a problemei se stabilește o formulă de recurență, pornind de la una din ecuațiile sistemului (3.242). Folosind a doua ecuație se obține:

$$\left([C]+\frac{2\Delta\tau}{3}[k]\right)\{t\}_{ej}=\left([C]-\frac{\Delta\tau}{3}[k]\right)\{t\}_{ei}+\Delta\tau\{f\}_j \tag{3.243}$$

Cu această relație de recurență se determină temperaturile $\{t\}_{ej}$ în nodurile elementului finit la momentul j în funcție de temperaturile din noduri $\{t\}_{ei}$ la momentul i precedent.

Ecuația elementului finit (3.243) se poate scrie sub formă concentrată, obișnuită:

$$[k]_i\{t\}_{ej}=\{f\}_i \tag{3.244}$$

unde:

$$[C]+\frac{2\Delta\tau}{3}[k]=[k]_i;\quad [C]-\frac{\Delta\tau}{3}[k]=[p];\quad [p]\{t\}_{ei}+\Delta\tau\{f\}_j=\{f\}_i \tag{3.245}$$

în care: $[k]_i$ este matricea de conductivitate a elementului în regim tranzitoriu; $\{f\}_i$ — vectorul încărcării termice în regim tranzitoriu.

Sistemul de ecuații (3.244) se constituie pentru toate elementele finite la fiecare pas de timp, și se asamblează în sistemul de ecuații al domeniului analizat, prin a cărui rezolvare se determină temperaturile nodale la pasul de timp j în funcție de valorile temperaturilor la momentul precedent, i.

Pentru rezolvarea cu metoda elementelor finite a problemelor de conducție termică în regim staționar s-a elaborat, pe baza modelului prezentat la punctul 3.8.6, programul ordinator ATEFS, compus din două unități ATEFS 1 și ATEFS 2, iar pentru calculul cîmpurilor conductive tranzitorii s-a întocmit, pe baza modelului descris la punctul 3.8.7, programul ordinator ATEFT. Acestea folosesc elementul finit liniar cu trei noduri. Programul ATEFS poate fi folosit fără nici o modificare și pentru studiul problemelor de curgere staționară în medii poroase.

3.9. APLICAREA METODEI REZIDUURILOR PONDERATE LA OBȚINEREA ECUAȚIILOR CU ELEMENTE FINITE PENTRU MIȘCAREA PERMANENTĂ A UNUI FLUID ÎNTRE DOI PEREȚI PLANI PARALELI

3.9.1. METODA REZIDUURILOR PONDERATE

Teoria reziduurilor ponderate permite extinderea formulărilor de tip variațional la probleme ce nu pot fi caracterizate printr-o funcțională.

Metoda reziduurilor ponderate este o metodă de obținere a soluțiilor aproximative pentru ecuații diferențiale liniare și neliniare. Prin această metodă se caută o soluție a problemei pe întregul ei domeniu de analiză.

Fie o problemă descrisă prin ecuația:

$$G(u)=0, \tag{3.246}$$

unde G este un operator general, iar ecuația (3.246) poate fi o ecuație diferențială ordinară, sau o ecuație diferențială cu derivate parțiale de tip parabolic sau hiperbolic.

Scanned with OKEN Scanner

Domeniul de analiză al problemei este D, suprafața care îl mărginește este s, iar condițiile inițiale și la limită pe s sînt de forma:

$$Q(u)=f(r,t) \tag{3.247}$$

în care Q este un operator diferențial, de ordin inferior lui G.

Se consideră soluția aproximativă de încercare:

$$\bar{u}(r,t)=u_0(r,t)+\sum_{i=1}^{n}\alpha_i(t)\cdot\varphi_i(r) \tag{3.248}$$

care se dorește convergentă spre soluția exactă $u(r,t)$ pe măsură ce $n\to\infty$. Această soluție aproximativă se alege astfel încît să satisfacă în mod exact condițiile inițiale și la limită ale problemei.

Pentru $i=1,2,\ldots,n$, funcțiile $\varphi_i(r)$ constituie un sistem complicat de funcții liniar independente pe domeniul D, iar $\alpha_i(t)$ sînt coeficienți ce urmează a fi determinați.

Se introduce soluția aproximativă (3.248) în ecuația (3.246) care nu mai este acum satisfăcută în mod exact:

$$G(\bar{u})=R(r,t,\alpha_i), \tag{3.249}$$

unde $R(r,t,\alpha_i)$ este un reziduu de calcul. Deoarece $R=0$ pentru soluția exactă, acest reziduu poate fi considerat ca măsură a erorii de aproximare.

Metoda reziduurilor ponderate constă în determinarea coeficienților α_i astfel încît eroarea R pe întregul domeniu considerat să fie cît mai mică. Aceasta se realizează prin formarea unei medii ponderate a erorii pe întregul domeniu și prin impunerea condiției:

$$\int_D R(r,t,\alpha_i)W_k(r)\,\mathrm{d}D=0\ (k=1,2\ldots,m) \tag{3.250}$$

în care $W_k(r)$ reprezintă un set de funcții de pondere liniar independente, ce depind de aceleași variabile independente ca și funcțiile $\varphi_i(r)$ din expresia (3. 248).

Pe măsură ce n crește progresiv, se așteaptă ca soluția aproximativă $\bar{u}$ să tindă spre soluția exactă u, iar reziduul R să devină tot mai mic. La limită cînd $n\to\infty$ se obține $\bar{u}\to u$ și $R\to 0$, iar ecuația (3.250) este satisfăcută pentru orice set de funcții de pondere $W_k(r)$.

Potrivit *procedeului Galerkin* de selectare a funcțiilor de pondere se poate scrie:

$$W_k(r)=\Phi_k(r) \tag{3.251}$$

unde $\Phi_k(r)$ sînt funcțiile de interpolare ale modelului cu elemente finite.

În această metodă s-a presupus că soluția de încercare (3.248) satisface în mod exact condițiile de frontieră și numai aproximativ ecuația diferențială generală. Se poate și invers, respectiv ca soluția propusă să satisfacă exact ecuația generală (3.246) și numai aproximativ condițiile de frontieră (3.247), caz în care se vorbește de *metode de frontieră*.

Scanned with OKEN Scanner

3.9.2. OBȚINEREA ECUAȚIILOR CU ELEMENTE FINITE

Pentru a ilustra aplicarea metodei Galerkin se consideră ca exemplu mișcarea permanentă a unui fluid între doi pereți plani paraleli (mișcarea Couette generalizată), al cărui model analitic este dat de ecuația:

$$\frac{dp}{dx}=\eta\frac{d^2v}{dy^2} \tag{3.252}$$

cu condițiile de frontieră:

$$\begin{aligned} &v=0, \text{ pentru } y=0\\ &v=v_0, \text{ pentru } y=y_0 \end{aligned} \tag{3.253}$$

în care: v este viteza fluidului; p — presiunea; η — vîscozitatea dinamică a fluidului.

Introducînd parametrii adimensionali $v_*=\frac{v}{v_0}$, $y_*=\frac{y}{y_0}$ și parametrul de presiune $p_*=\frac{y_0^2}{\eta v_0}\left(-\frac{dp}{dx}\right)$, se obține ecuația:

$$\frac{d^2v_*}{dy_*^2}+p_*=0 \tag{3.254}$$

cu condițiile:

$$\begin{aligned} &v_*=0, \text{ pentru } y_*=0\\ &v_*=1, \text{ pentru } y_*=1 \end{aligned} \tag{3.255}$$

Pentru discretizare se folosesc elemente finite unidimensionale cu cîte trei noduri. Alegînd la nivelul unui element coeficienții de pondere $W_k=N_k$ ($k=1, 2, 3$) și aplicînd condiția generală (3.250) se obține:

$$\int_{L^e} N_k\left(\frac{d^2\bar{v}}{dy_*^2}+p_*\right)dy_*=0 \tag{3.256}$$

unde N_k reprezintă funcțiile de formă.

Înlocuind primul termen al integrantului prin expresia:

$$N_k\frac{d^2\bar{v}}{dy_*^2}=\frac{d}{dy_*}\left(N_k\frac{d\bar{v}}{dy_*}\right)-\frac{dN_k}{dv_*}\frac{d\bar{v}}{dy_*} \tag{3.257}$$

și aplicînd condițiile la limită se obține:

$$\int_{L^e}\frac{dN_k}{dy_*}\frac{d\bar{v}}{dy_*}dy_*=\int_{L^e}N_kp_*dy_* \tag{3.258}$$

Considerînd pe rînd indicii $k=1, 2, 3$ se obține ecuația caracteristică a elementului generic e, sub formă matriceală:

$$A^eV^e=B^e \tag{3.259}$$

în care:

$$A^e_{mn}=\int\frac{dN_m}{dy_*}\frac{dN_n}{dy_*}dy_*; \qquad B^e_m=\int p_*N_m\,dy_* \tag{3.260}$$

Scanned with OKEN Scanner

Asamblarea elementelor finite pe întreg domeniul de analiză conduce în final la ecuația matriceală:

$$A\ V = B \tag{3.261}$$

în care: A este matricea de rigiditate a sistemului; B — vectorul încărcării. Elementele acestor matrice se obțin astfel:

$$A_{mn} = \sum_{e=1}^{2} A_{mn}^{e}\ ; \qquad B = \sum_{e=1}^{2} B_{m}^{e} \tag{3.262}$$

Ecuația (3.261) împreună cu condițiile la limită impuse nodurilor de frontieră:

$$\begin{aligned} v_1 &= 0, \text{ pentru } y_* = 0 \\ v_9 &= 0, \text{ pentru } y_* = 1, \end{aligned} \tag{3.263}$$

constituie modelul analog cu elemente finite al problemei, a cărui soluționare conduce la obținerea valorilor nodale de viteză $v_1, v_2, \ldots, v_9$. Orice valoare intermediară se obține apoi pe baza acestor valori cu ajutorul funcțiilor de formă corespunzătoare tipului de elemente considerate.

Scanned with OKEN Scanner

4

Metode ale calculului operațional aplicabile la optimizarea proiectării instalațiilor

4.1. PRELIMINARII MATEMATICE PRIVIND FORMULAREA UNEI PROBLEME DE OPTIMIZARE

Necesitatea creșterii similitudinii dintre procesele reale din instalațiile pentru construcții și modelele matematice care le reprezintă conduce inevitabil la sporirea complexității modelelor și implicit la imposibilitatea rezolvării lor fără aplicarea unor metode și tehnici moderne de calcul și utilizarea calculatoarelor electronice.

În acest context, disciplinele matematice moderne prin calculul operațional pun la îndemîna proiectantului un vast aparat de analiză științifică în stabilirea deciziilor optime pentru problemele proiectării instalațiilor hidraulice, termice și electrice. O mare contribuție în proiectarea optimală a acestor instalații o aduce crearea programării matematice (liniară, neliniară, dinamică, geometrică etc.) și a teoriei grafurilor, lărgindu-se aria de modelare a problemelor optimizării.

Proiectarea unei anumite instalații poate fi făcută în mai multe alternative, în funcție de criteriile care stau la bazele calculului. Oricare ar fi aceste criterii, rezultatul proiectării conduce la alegerea unor valori pentru variabilele de proiectare care pot fi mărimi de natură fizică diferită cum ar fi lungimi, debite, presiuni, temperaturi, curenți etc. Proiectarea optimală determină variabilele de proiectare ce respectă anumite condiții (restricțiile proiectării) impuse de comportarea și funcționalitatea instalației, minimizînd o funcție de eficiență (funcția obiectiv).

Pentru rezolvarea problemei de optimizare este necesară cunoașterea unor metode adecvate de calcul.

Metodele aplicate pentru determinarea soluției optime nu permit în general obținerea unei soluții calitativ diferite pentru problema în studiu, ci conduc la alegerea celor mai bune valori, adică optimizează soluția în concepția de proiectare a utilizatorului.

Optimizarea unui sistem presupune următoarele etape principale:

— delimitarea cadrului problemei;
— definirea criteriului de optimizare;
— găsirea relațiilor tehnologice care descriu limitările sistemului;
— rezolvarea programului matematic de optimizare.

Scanned with OKEN Scanner

Procesul de optimizare se poate încheia printr-o ultimă analiză suplimentară care să țină seama de elemente necuantificabile: sociale, ecologice, politice etc., ce pot deplasa optimul matematic.

4.1.1. VARIABILELE DE PROIECTARE

Mărimile geometrice, hidraulice, termice, economice care urmează să fie calculate la rezolvarea unei probleme de proiectare se numesc *variabile de proiectare*. Spre exemplu, la proiectarea optimală a rețelelor de apă, variabilele de proiectare sînt diametrele tronsoanelor, debitele pe fiecare tronson, parametrii ce descriu un anumit regim de curgere a apei, acestea grupîndu-se în clase de variabile între care există o strînsă interdependență.

În cazul rețelelor inelare se stabilește geometria rețelei și debitul de calcul ce încarcă rețeaua dintr-o anumită ipoteză de funcționare. Ca necunoscute apar atît diametrele, cît și debitele sau pierderile de sarcină pe fiecare tronson, variabile dependente unele de altele.

În rezolvarea clasică se predimensionează secțiunea tronsoanelor (diametrele) pe considerentul vitezelor economice de curgere a apei, după care se calculează debitele reale pe fiecare tronson, în condițiile respectării ecuațiilor de compatibilitate a rețelei.

Prin optimizarea matematică a rețelei inelare, în formularea criteriului de optimizare se poate alege ca și clasă de variabile fie debitele sau pierderile de sarcină, fie diametrele, stabilindu-se dependența dintre ele prin restricțiile matematice rezultate din condițiile hidraulice și economice.

4.1.2. RESTRICȚIILE DE PROIECTARE

Restricțiile reprezintă un ansamblu de relații matematice care descriu cît mai fidel interdependența variabilelor proceselor ce au loc în cadrul sistemului de optimizat. Acestea pot exprima direct anumite limitări impuse unei variabile sau unui grup al variabilelor (restricții explicite) sau pot exprima indirect aceste limitări asupra unor mărimi de care depind variabilele (restricții implicite).

Toate restricțiile se pot transforma în egalități sau inegalități mai mult sau mai puțin complexe.

Egalitățile din cadrul acestor restricții provin din legi generale sau specifice, care impun numai anumite legături între parametrii decizionali și din ecuații de conservare: masă (debit), energie (putere) sau impuls.

Inegalitățile semnifică imposibilitatea depășirii unor anumite limite care constituie, de obicei, minimele sau maximele tehnice ale sistemului. Totodată, situațiile în care se impune asigurarea obligatorie a unor mase, debite, puteri etc. pot genera restricții de tip inegalitate.

4.1.3. FUNCȚIA OBIECTIV

Funcția obiectiv exprimă relația matematică dintre variabilele de proiectare și criteriile de optimizare adoptate, ce trebuie optimizată (minimizată, maximizată).

Valoarea funcției obiectiv, de obicei se limitează printr-o serie de restricții, avînd de-a face în această situație cu o optimizare condiționată, iar în cazul că restricțiile lipsesc optimizarea este necondiționată.

Scanned with OKEN Scanner

Funcția obiectiv poate fi concepută astfel încît să exprime criterii simple de cost, consumuri energetice, consumuri de materiale deficitare etc. sau să exprime mai multe criterii simultan (funcție multicriterială).

În funcție de scopul urmărit, aceste criterii se pot împărți în patru categorii: — tehnice; — economice; — energetice; — tehnico-economice.

Criteriile tehnice sînt cele care iau în considerare doar efectele elementelor tehnologice. Din cadrul acestora, cele mai răspîndite sînt: randamente, consumuri și respectiv consumuri specifice.

Criteriile economice cuprind cele două mari categorii: investiții și cheltuieli de exploatare.

Estimarea investițiilor sau a investițiilor specifice pentru unele echipamente sau utilaje (compresoare, conducte, evaporatoare, generatoare de abur, pompe, schimbătoare de căldură etc.) se poate realiza folosind ecuații de corelare geometrică de forma (2.8), în care coeficientul c_0 ține seama de investițiile suplimentare datorate construcției și montajului, pe lîngă cele efective.

Cheltuielile de producție se consideră, de obicei, pentru o perioadă de un an. Ele cuprind cheltuielile efective și cele de echivalare.

Cheltuielile efective se compun din:

— cheltuieli pentru exploatarea, întreținerea și repararea utilajelor și instalațiilor;

— amortizări ale investițiilor.

Cheltuielile de echivalare iau în considerare aducerea variantelor la echivalență din punct de vedere al efectului lor util.

Atunci cînd se analizează variantele și din punct de vedere fiabilistic este necesar să se ia în considerare și daunele probabile, care reprezintă valoarea pagubei probabile produse prin degradare de materiale sau materii prime sau valoarea producției nerealizate ca urmare a întreruperilor accidentale în funcționarea sistemului.

Criteriile energetice cuprind energia înglobată în elementele sistemului și energia consumată pentru exploatarea sistemului.

Criteriile tehnico-economice sînt criterii complexe care exprimă atît aspectul tehnologic, cît și cel economic, cele mai utilizate fiind: cheltuielile anuale echivalente, cheltuielile totale actualizate, termenul de recuperare al investiției.

4.1.4. MODELUL MATEMATIC AL UNEI PROBLEME DE OPTIMIZARE

Orice model de optimizare presupune crearea unei funcții obiectiv care trebuie optimizată.

Fie $X \in D$, $D \subset R$; $X=\{x_1 x_2 \ldots x_n\}$ — vectorul variabilelor de proiectare și F: $D \to R$, $F(X)=F(x_1 x_2, \ldots, x_n)$ — funcția obiectiv.

Se consideră funcțiile $\varphi_i : D \to R$; $\varphi_i(X)=\varphi_i(x_1, x_2, \ldots x_n)$, $i=1,\ldots,m$, care caracterizează restricțiile proiectării, sub forma:

$$\begin{aligned} &\text{— restricții funcționale: } \varphi_i(X)=0;\ i=1,2,\ldots,p \\ &\text{— restricții regionale: } \varphi_j(X)<S_j;\ j=p+1,\ldots,m \end{aligned} \qquad (4.1)$$

Dacă $\Omega \subset D$ este mulțimea soluțiilor sistemului de restricții (4.1) numite soluții admisibile, problema de optimizare este:

$$\{F(X) \mid X \in \Omega\} \to \min \ (\max) \qquad (4.2)$$

Scanned with OKEN Scanner

În figura 4.1 se reprezintă grafic, spre exemplificare, următoarea problemă de optimizare:

$$F(x_1, x_2) \to \min \tag{4.3}$$

$$\varphi_1(x_1, x_2)=0; \quad s_2 \leqslant \varphi_2(x_1, x_2) \leqslant S_2; \quad s_3 \leqslant \varphi_3(x_1, x_2) \leqslant S_3,$$

unde x_1 și x_2 formează planul x_1Ox_2, iar axa OF se ia perpendiculară pe acest plan. Funcția $F(X)$ se prezintă ca o suprafață, numită *suprafața criteriu* în spațiul tridimensional.

Restricția funcțională $\varphi_1(x_1, x_2)=0$ descrie o curbă în planul bazei, care proiectată pe suprafața criteriu formează o curbă în spațiu. Un punct parcurgînd această curbă în spațiu urcă sau coboară, funcție de suprafața $F(x_1, x_2)$.

Restricțiile regionale $\varphi_2(x_1, x_2)$ și $\varphi_3(x_1, x_2)$ reprezintă două familii de curbe plane, extremele lor putînd fi desenate între limitele inferioare și superioare s_i, respectiv $S_i (i=2, 3)$.

Prin proiectarea regiunii cuprinse între curbele $\varphi_2=s_2$ $\varphi_2=S_2$; $\varphi_3=s_3$, $\varphi_3=S_3$ se obține regiunea admisibilă pe suprafața criteriu. Cea mai mică poziție a unui punct în mișcare pe curba $\varphi_1=0$, aparținînd acestui domeniu admisibil corespunde optimului funcției obiectiv.

Pentru găsirea soluțiilor tehnice optime, adică a condițiilor corespunzătoare minimului sau maximului unei funcții obiectiv, aceasta trebuie cercetată cu ajutorul unor metode de optimizare analitică sau numerică descrise în numeroase lucrări de specialitate [21], [40], [46], [58], [95], [102], în această lucrare expunîndu-se, atît pentru optimizarea necondiționată, cît și pentru optimizarea condiționată, doar acele metode care se folosesc cu precădere în cadrul unor modele de calcul optimal al instalațiilor.

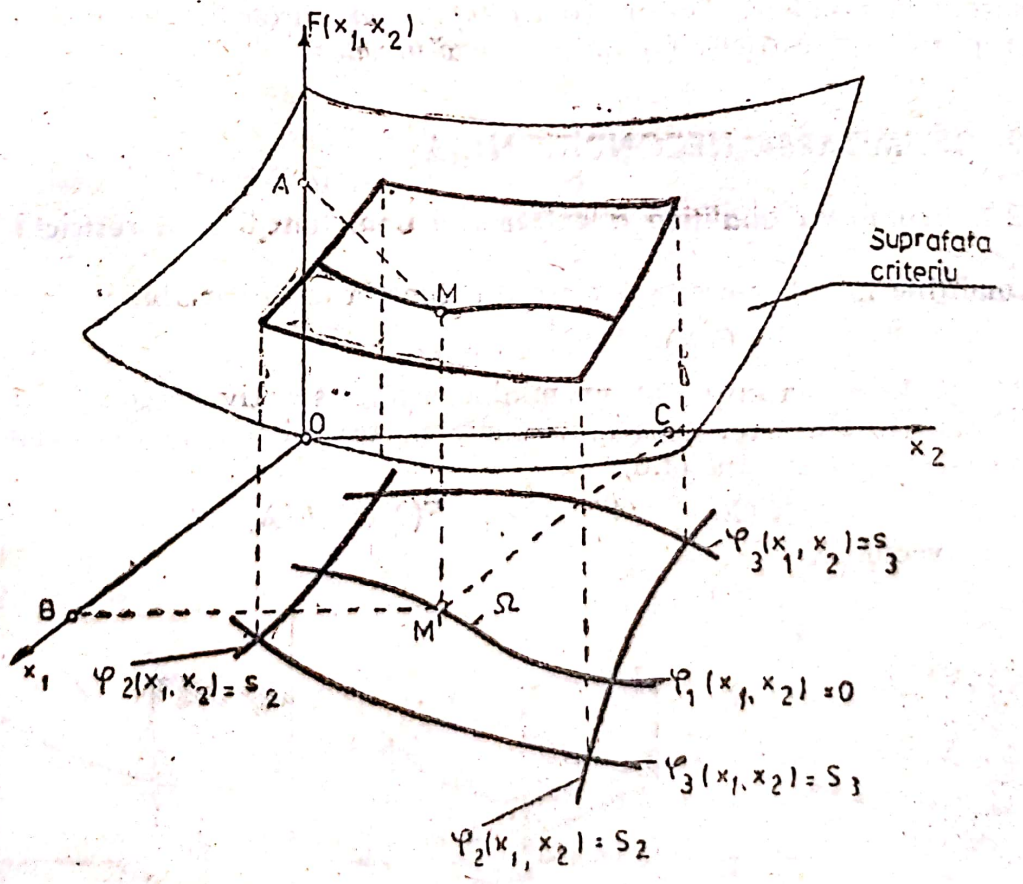


Fig. 4.1. Reprezentarea grafică a unei probleme de optimizare

Scanned with OKEN Scanner

4.2. ELEMENTE DE PROGRAMARE MATEMATICĂ

4.2.1. NOȚIUNI GENERALE

Programarea matematică a apărut ca o metodă de optimizare, deci de aflare a extremului unei funcții ale cărei argumente sînt supuse la anumite restricții. Este vorba în principal de o funcție scalară a cărei mulțime de definire este mărginită, ceea ce conduce la faptul că extremul constrîns poate fi diferit de cel liber.

Spre exemplificare, se consideră cazul minimului funcției $F(x)$ al cărui argument x este supus restricției $x \geqslant a$, $a \in R$ (fig. 4.2).

Se observă că în primele două situații (fig. 4.2, *a*, *b*) minimul constrîns este un minim analitic:

$$\left.\frac{\mathrm{d}F(x)}{\mathrm{d}x}\right|_m = 0 \tag{4.4}$$

punctul x_m găsindu-se în primul caz în interiorul mulțimii $x \geqslant a$, iar în cel de-al doilea caz pe frontiera acestei mulțimi.

În cel de-al treilea caz, minimul constrîns nu este analitic:

$$\left.\frac{\mathrm{d}F(x)}{\mathrm{d}x}\right|_m = 0, \tag{4.5}$$

nefiind situat pe frontiera mulțimii $x \geqslant a$. Acest ultim caz se încadrează efectiv în domeniul programării matematice, întrucît rezolvarea lui necesită metode specifice.

Cele trei cazuri prezentate anterior exemplifică și gruparea problemelor de optimizare în probleme fără restricții (*optimizare necondiționată*) și respectiv probleme cu restricții (*optimizare condiționată*).

4.2.2. OPTIMIZAREA NECONDIȚIONATĂ

4.2.2.1. Evaluarea analitică a extremului unei funcții fără restricții

În condițiile în care nu există restricții, funcția de n variabile:

$$F(X) = F(x_1, x_2, \ldots, x_n) \tag{4.6}$$

cu derivate parțiale continue are un minim local, respectiv maxim local în punctul caracterizat de X_m, dacă în vecinătatea acestui punct este îndeplinită condiția (4.7), respectiv (4.8):

$$F(X_m) \leqslant F(X) \tag{4.7}$$

$$F(X_m) \geqslant F(X) \tag{4.8}$$

pentru orice vector X.

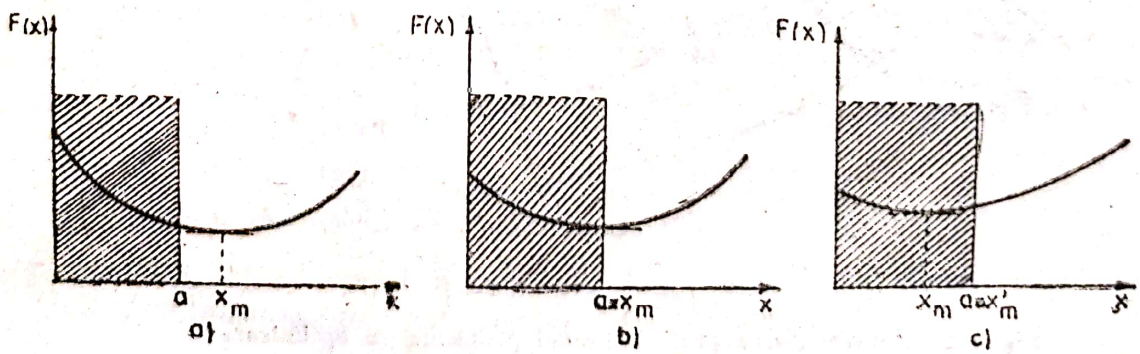


Fig. 4.2. Graficul funcției $F(x)$

Scanned with OKEN Scanner

O condiție necesară și suficientă pentru ca o funcție derivabilă $F(X)$ să aibă un minim local unic în punctul X îl constituie satisfacerea relațiilor:

$$\frac{\partial F(X)}{\partial x_i}=0 \quad (i=1,2,\ldots,n) \tag{4.9}$$

respectiv matricea hessian:

$$H=\begin{vmatrix} \frac{\partial^2 F(X)}{\partial x_1^2} & \frac{\partial^2 F(X)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 F(X)}{\partial x_1 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 F(X)}{\partial x_n \partial x_1} & \frac{\partial^2 F(X)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 F(X)}{\partial x_n^2} \end{vmatrix} \tag{4.10}$$

să fie pozitiv definită.

Se știe că o condiție necesară și suficientă pentru ca o matrice nedegenerată să fie pozitiv definită este ca toți minorii principali ai matricei să fie pozitivi.

Orice punct interior de extrem local în care funcția F este diferențiabilă și diferențiala sa este nulă în acest punct, reprezintă un *punct staționar* al funcției. Există puncte staționare care nu sînt puncte de extrem, ce se numesc *puncte „șa"* ale lui F. La funcțiile de o singură variabilă, punctele în care derivata întîi se anulează, dar care nu sînt puncte de extrem, sînt *puncte de inflexiune*.

Condițiile (4.9) și (4.10) sînt necesare pentru determinarea unui minim local unic, care nu întotdeauna reprezintă cel mai mic minim, adică minimum global.

Un extrem relativ (local) este și un extrem absolut (global) dacă funcția F este convexă.

4.2.2.2. Domenii și funcții convexe

Un domeniu deschis D mărginit de $\varphi_j(X)\leqslant 0$ $(j=1,2,\ldots,m)$ se numește *convex* dacă orice segment de dreaptă $\mu X_1+(1-\mu)X_2$; $(0<\mu<1)$ care unește două puncte caracterizate de vectorii de poziție $X_1\in D$, $X_2\in D$ este conținut în D.

În figura 4.3 se exemplifică un domeniu convex (fig. 4.3., *a*) și un domeniu neconvex (fig. 4.3, *b*).

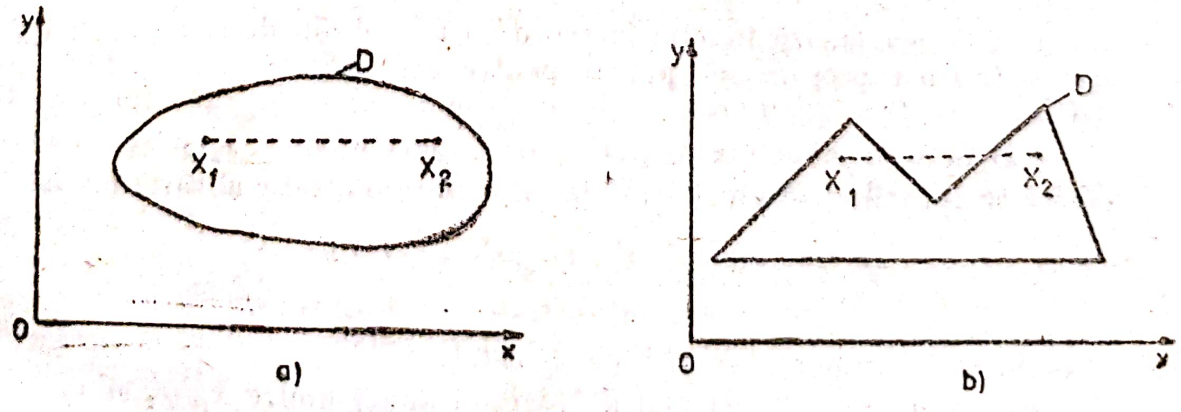


Fig. 4.3. Reprezentarea grafică a unor domenii convexe și neconvexe

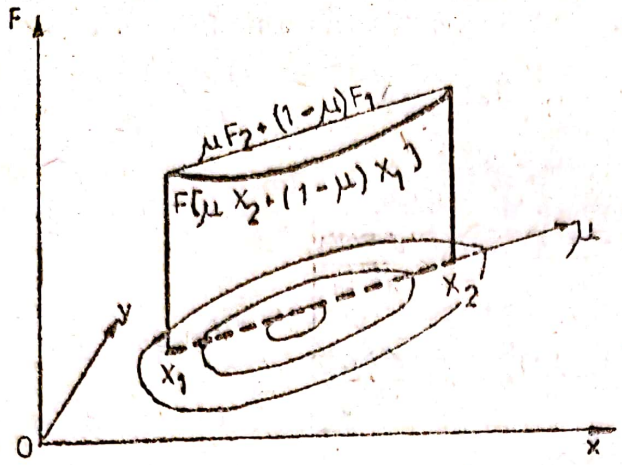


Fig. 4.4. Reprezentarea grafică a unei funcții convexe

O funcție ($F(X)$ este convexă (fig. 4.4) dacă pe orice dreaptă care unește două puncte X_1, $X_2 \in D$ este îndeplinită condiția:

$$F[\mu X_2+(1-\mu)X_1] \leqslant \mu F(X_2)+(1-\mu)F(X_1);\ \forall\mu\in(0,1) \quad (4.11)$$

unde partea dreaptă a inegalității reprezintă interpolarea liniară a funcției de-a lungul dreptei ce unește pe X_1 cu X_2.

Funcțiile convexe au unele proprietăți importante:

a) $F(X_1) \geqslant F(X_2) + (X_1 - X_2)\nabla F(X_2)$, (4.12)

pentru toți X_1, X_2;

b) matricea derivatelor parțiale $\partial^2 F/\partial x_i \partial x_j$ este pozitiv semidefinită pentru toate punctele X, adică are și valori caracteristice pozitive și negative;

c) fiecare minim local într-un domeniu convex D reprezintă un minim global pentru acest domeniu.

4.2.2.3. Metode numerice de evaluare a optimului unei funcții fără restricții

Metodele analitice bazate pe calculul diferențial au o utilizare relativ limitată la rezolvarea problemelor de optim, datorită complexității funcției obiectiv. De aceea se apelează la unii algoritmi de optimizare numerică a funcțiilor.

Metoda de evaluare a minimului unei funcții monovariabile prin interpolare. Metoda de interpolare constă, de regulă, în aproximarea funcției printr-un polinom de interpolare pătratică sau cubică și cuplată cu un algoritm de căutare cu pas constant sau accelerat; poate fi folosită la evaluarea extremului unei funcții monovariabile sau ca etapă de evaluare a extremului după o direcție, în cadrul unui algoritm de determinare a extremului unei funcții multivariabile.

Un algoritm de căutare a minimului unei funcții monovariabile, care cuplează căutarea cu pas accelerat cu interpolarea pătratică, cuprinde următoarele etape:

E1) Se cercetează funcția în acord cu algoritmul de căutare cu pas accelerat într-o succesiune de puncte pentru care rezultă: $F(x_1)>F(x_2)>\ldots>F(x_{n-1})$ și $F(x_{n-1})<F(x_n)$, x_n fiind primul punct în care funcția crește.

E2) Se reține punctele x_{n-2}, x_{n-1} și x_n, notate cu x_1, x_2 și x_3, cu ajutorul cărora se aproximează funcția printr-un polinom pătratic al cărui minim:

$$r_* = \frac{1}{2}\,\frac{F(x_1)(x_3^2-x_2^2)+F(x_2)(x_1^2-x_3^2)+F(x_3)(x_2^2-x_1^2)}{F(x_1)(x_3-x_2)+F(x_2)(x_1-x_3)+F(x_3)(x_2-x_1)} \quad (4.13)$$

aproximează minimul funcției de cercetat.

E3) Se înlocuiește cel mai nefavorabil punct dintre x_1, x_2 și x_3 cu r_* și se repetă etapa E2.

Scanned with OKEN Scanner

După a doua interpolare, și în continuare, se verifică dacă interpolarea este reușită, adică dacă funcția în noul punct de interpolare este mai favorabilă decît în punctul din interpolarea anterioară. Se declară, de asemenea, interpolare nereușită cazul cînd numitorul expresiei (4.13) devine egal cu zero.

Dacă interpolarea este reușită, se înlocuiește cel mai nefavorabil punct dintre x_1, x_2 și x_3 cu r_* și se trece la etapa următoare E3.

Dacă interpolarea nu este reușită, se acceptă drept punct de start punctul de interpolare anterioară reușită, se reduce pasul și căutarea se reia de la etapa E1.

E4) Se verifică criteriul de convergență, cum ar fi diferența dintre valorile funcției în două puncte de interpolare reușită consecutive, să fie mai mică decît o valoare critică ε. Dacă convergența nu este satisfăcută, se reia algoritmul de la etapa E2.

În figura 4.5. se prezintă schema logică a programului ordinator FMIN 1, elaborat pe baza acestui algoritm, unde $X(1)$este punctul de start; P — pasul; R — coeficientul de reducere a pasului în cazul unei interpolări nereușite; ε — valoarea critică de convergență; X_0 — rezultatul de interpolare.

În unele situații interpolarea printr-o formă pătratică a funcției de minimizat pe intervalul care conține minimul nu este suficient de precisă. În consecință, s-au propus polinoame de interpolare de ordin superior, dintre care polinomul de gradul trei este cel mai utilizat.

Alte metode directe de evaluare a extremului unei funcții monovariabile sînt metodele UNIPLEX [58] și BRENT [11], [21], pe baza ultimei elaborîndu-se programul ordinator FMIN 2.

Metode de evaluare a minimului unei funcții multivariabile.

A. *Clasificarea și descrierea principalelor tipuri de metode*

Metodele numerice de determinare a minimului unei funcții multivariabile fără restricții pot fi împărțite în trei categorii:

— metode aleatoare de căutare;
— metode de căutare unidirecțională;
— metode de gradient.

Primul tip de metode conduce la aproximarea minimului global, iar celelalte două metode conduc, în general, doar la determinarea unui minim local.

Metodele aleatoare de căutare au la bază generarea unor numere aleatoare într-un anumit domeniu, numere care se atribuie componentelor vectorului $X=\{x_1x_2...x_n\}$, urmînd a calcula funcția $F(X)$ pentru fiecare grup de valori date componentelor. Luînd un număr mare de încercări, se consideră ca valoare minimă a funcției $F(X)$, cea mai mică valoare calculată.

Metodele de căutare unidimensională se bazează pe modificarea succesivă a componentelor vectorului X. În acest sens, se pornește cu modificarea primei componente x_1, pînă cînd funcția $F(X)$ nu mai scade. Apoi se trece la a doua componentă x_2 ș.a.m.d., pînă ce funcția $F(X)$ a atins minimul căutat cu precizia impusă. Din categoria metodelor de căutare directă multivariabile, cele mai utilizate sînt metoda Hooke-Jevees, metoda Powel și metoda Rosenbrook [58], [70].

Metodele de gradient efectuează modificări simultane ale tuturor variabilelor de care depinde funcția $F(X)$, astfel ca aceste modificări să conducă la o deplasare în spațiul n dimensional, într-o direcție opusă gradientului funcției obiectiv, care reprezintă vectorul ortogonal la conturul lui $F(X)$ ce trece prin punctul de coordonate X.

Scanned with OKEN Scanner

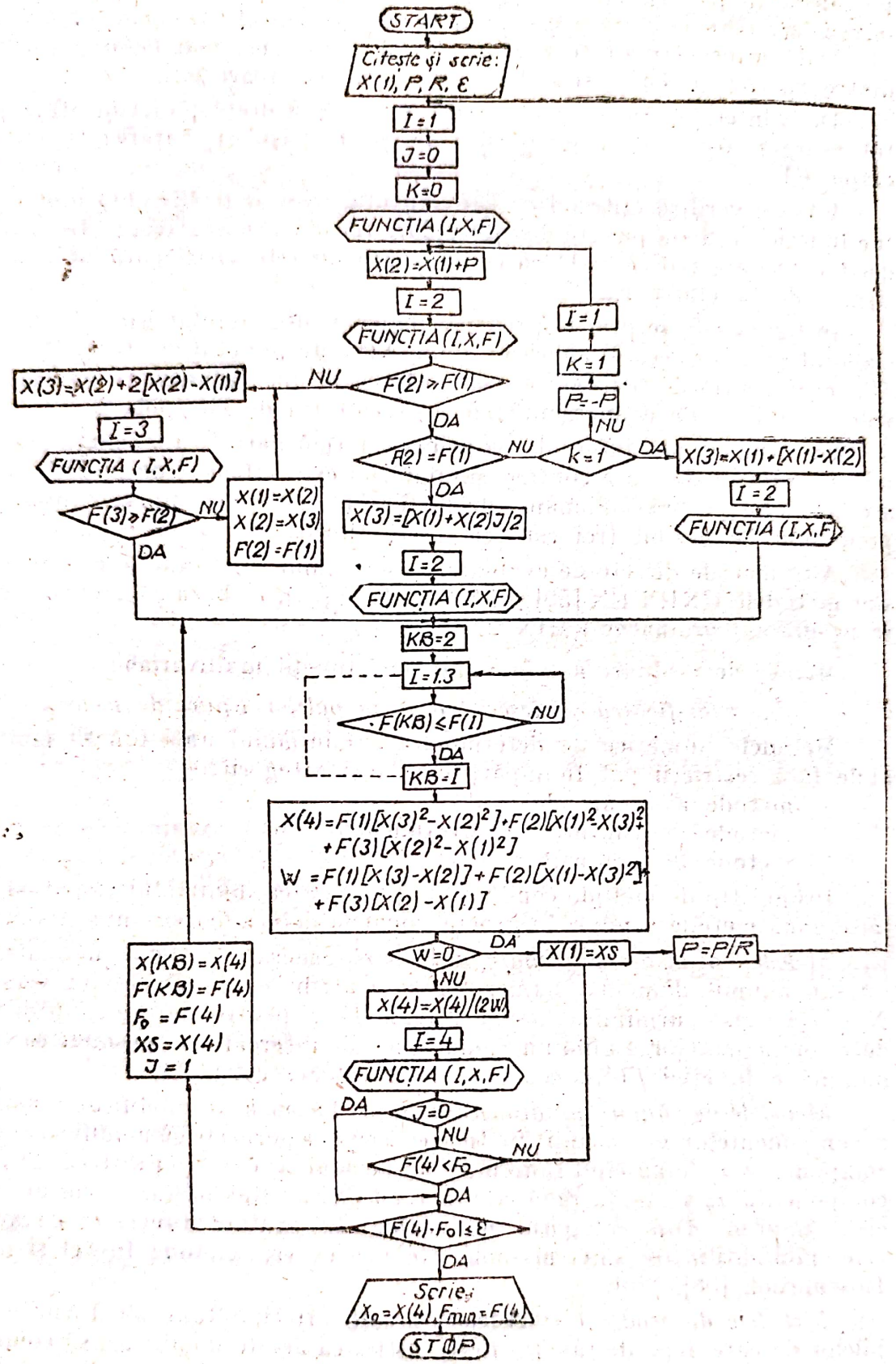


Fig. 4.5. Schema logică a algoritmului de căutare cu pas accelerat cuplat cu interpolare pătratică pentru determinarea minimului unei funcții monovariabile

Scanned with OKEN Scanner

Întrucît dintre metodele menționate cel mai frecvent se utilizează cele din ultima categorie, în continuare se prezintă unele dintre acestea.

B. *Metode de gradient*

Cele mai eficiente metode de evaluare a minimului unei funcții multivariabile au la bază informațiile locale pentru fiecare punct, date prin gradient. Această categorie de metode aplicabile la optimizarea funcțiilor diferențiabile, se bazează pe ideea că variația cea mai pronunțată a unei funcții $F(X)$ se obține prin deplasarea după direcția gradientului. După cum funcția obiectiv este aproximată la dezvoltarea în serie Taylor printr-o expresie liniară sau pătratică, aceste metode se clasifică în metode de gradient de ordinul 1, sau metode de gradient de ordinul 2.

a) *Metoda gradientului simplu.* Considerînd o funcție obiectiv $F(X)$ de n variabile, gradientul în punctul X este vectorul derivatelor de ordinul *1* în acest punct:

$$G=\nabla F(X)=\left\{\frac{\partial F}{\partial x_1}\ \frac{\partial F}{\partial x_2}\ \ldots\ \frac{\partial F}{\partial x_n}\right\} \tag{4.14}$$

Vectorul unitate (direcțiile normate ale gradientului) corespunzător punctului X_k are expresia:

$$D_k=\pm\frac{\nabla F(X_k)}{\|\nabla F(X)_k\|}=\pm\frac{G_k}{\|G_k\|} \tag{4.15}$$

în care norma euclidiană a vectorului G_k este:

$$\|G_k\|=[G_k^T G_k]^{1/2}=\left[\sum_{i=1}^{n}\left(\frac{\partial F}{\partial x_i}\right)_k^2\right]^{1/2} \tag{4.16}$$

iar semnul plus se utilizează pentru maximizare și semnul minus pentru minimizare.

Dacă se face referire la o problemă de minimizare, algoritmul cuprinde următorii pași:

P1) Se alege un punct de start X_0.

P2) Se evaluează gradientul de ordinul 1, $G_0=\nabla F(X_0)$ și direcțiile normate corespunzătoare D_0, în acest punct.

P3) Se evaluează minimul $X_1=X_0-r_* D_0$ după prima direcție ce trece prin X_0 determinînd parametrul r_* din condiția de minimizare a funcției $F(X_0+r_* D_0)$, punct utilizat drept vîrf de start la iterația următoare. Generalizînd, pentru fiecare punct de plecare definit de vectorul X_k se determină un nou punct definit de vectorul X_{k+1} pentru care funcția F are cea mai mare descreștere, conform cu relația:

$$X_{k+1}=X_k-r_* D_k, \tag{4.17}$$

relația concretă de determinare a parametrului r_* putînd fi (4.13).

P4) După fiecare iterație k se verifică criteriul de convergență:

$$\|G_k\|\leqslant\varepsilon \tag{4.18}$$

unde valoare a lui ε este legată de precizia cu care se determină minimul.

Dacă criteriul este satisfăcut, cercetarea se oprește și se admite drept minim al funcției obiectiv minimul ultimei direcții, iar în caz contrar se recurge la o nouă iterație.

Scanned with OKEN Scanner

În consecință, procesul de minimizare printr-o metodă de gradient implică repetarea a două etape principale:

— determinarea direcției de deplasare;

— alegerea valorii parametrului r_* care minimizează funcția într-o anumită direcție.

Prima etapă reprezintă elementul distinctiv pentru fiecare metodă de gradient, pe cînd cea de-a doua etapă este comună și similară metodei de căutare unidimensională.

Principalul dezavantaj al metodelor de gradient simplu îl constituie convergența relativ lentă în apropierea optimului, ajungîndu-se la o mișcare ciclică în jurul soluției, fără posibilitatea de a o atinge la anumite valori ale preciziei.

b) *Metoda gradienților conjugați* înlătură dificultățile de convergență ale metodei de gradient simplu, utilizînd pentru calculul unei direcții de explorare formula Fletcher-Reeves:

$$D_{k+1} = -G_{k+1} + \beta_k D_k; \quad k \geqslant 1 \tag{4.19}$$

în care coeficientul de ponderare β_k este dat de relația:

$$\beta_k = \frac{\|G_{k+1}\|^2}{\|G_k\|^2} \tag{4.20}$$

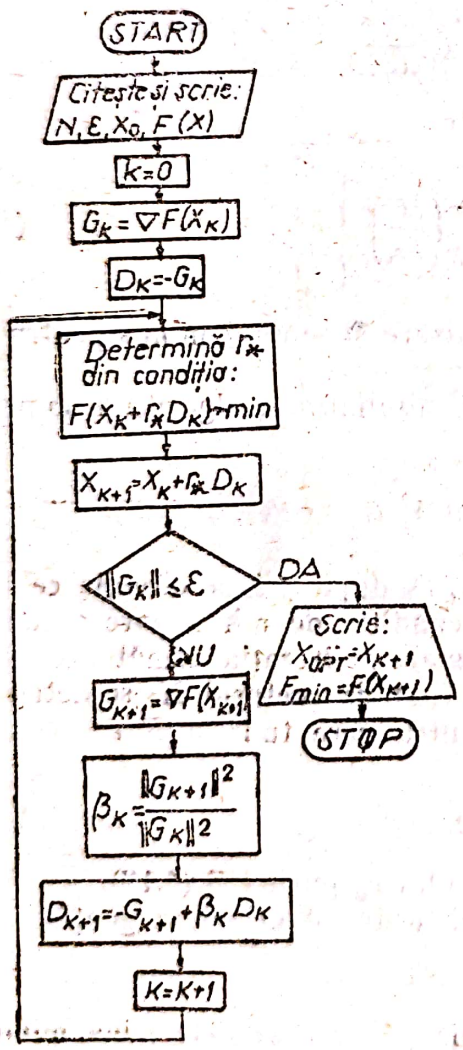


Fig. 4.6. Schema logică a algoritmului gradienților conjugați pentru determinarea minimului unei funcții multivariabile

Această metodă conduce la algoritmul cuprinzînd următorii pași:

P1) Se alege un punct inițial caracterizat de vectorul de poziție X_0.

P2) Se calculează gradientul funcției:

$$G_0 = \nabla F(X_0) \tag{4.21}$$

și se stabilește direcția inițială:

$$D_0 = -G_0 \tag{4.22}$$

P3) Se calculează gradientul funcției în X_{k+1}:

$$G_{k+1} = \nabla F(X_{k+1}) \tag{4.23}$$

urmînd a determina coeficientul β_k cu relația (4.20), iar apoi se stabilește noua direcție:

$$D_{k+1} = -G_{k+1} + \beta_k D_k \tag{4.24}$$

Dacă condiția de convergență (4.18) nu este îndeplinită, se revine la pasul P3, iar în caz contrar se admite drept minim al funcției obiectiv minimul ultimei direcții.

În figura 4.6. se prezintă schema logică a programului ordinator FMIN 3, elaborat pe baza acestui algoritm.

Algoritmul descris nu utilizează derivatele de ordinul 2 și inversări de matrice și prin urmare este ușor accesibil din punct de vedere practic. Prezintă avantajul unui volum relativ redus de calcule și necesită memorie puțină.

Scanned with OKEN Scanner

Dificultatea principală în utilizarea metodei gradienților conjugați este necesitatea folosirii unei căutări liniare precise pentru a se garanta convergența superliniară. În plus, dacă lungimea pasului nu este suficient de apropiată de cea optimală, direcția generală (4.19) poate să nu fie o direcție de descreștere. Remediul tipic în acest caz este reinițializarea algoritmului cu $D_k = -G_k$.

Metoda gradienților conjugați se folosește în special pentru rezolvarea problemelor de dimensiuni mari (peste 100...150 variabile).

Altă metodă de gradient este metoda Davidon-Fletcher-Powel [21], [58], ce face parte din categoria metodelor cvasinewtoniene și este preferată pentru probleme cu un număr mic sau mediu de variabile.

4.2.3. OPTIMIZAREA CONDIȚIONATĂ

4.2.3.1. Metode analitice de evaluare a extremului unei funcții cu restricții

S-a arătat la pct. 4.1.4. că o problemă de optimizare neliniară cu p restricții de egalitate și $m-p$ restricții de tip inegalitate, se formulează astfel:

$$\begin{aligned} &F(X)=F(x_1, x_2,\ldots,x_n)\to \min\ (\max)\\ &\varphi_i(X)=0;\ i=1,2,\ldots,p \\ &\varphi_j(X)\geqslant 0;\ j=p+1,\ldots,m \end{aligned} \tag{4.25}$$

Deoarece oricare din inegalități poate fi transformată într-o egalitate, prin introducerea unei variabile w_j:

$$\varphi_j(X)-w_j^2=0, \tag{4.26}$$

problema (4.25) devine o problemă de optimizare cu restricții de tip egalitate. O astfel de problemă se poate rezolva fie prin *metoda substituției directe*, fie prin *metoda multiplicatorilor Lagrange*.

Metoda substituției directe constă din explicitarea a $q\leqslant p$ variabile (dacă aceasta este posibil) și substituirea lor în funcția obiectiv, problema rezolvîndu-se în continuare ca una fără restricții. În unele cazuri practice explicitarea de variabile în cadrul restricțiilor de tip egalitate nu este posibilă.

În cazul căutării extremului unei funcții $F(X)$ de n variabile cu m restricții de tip egalitate ($m<n$), cu ajutorul multiplicatorilor Lagrange, se demonstrează că punctul X_0, extremul funcției obiectiv, se obține prin optimizarea funcției fără restricții:

$$\Gamma(X,\Lambda)=\Gamma(x_1, x_2,\ldots,x_n, \lambda_1, \lambda_2,\ldots,\lambda_m)=F(x_1, x_2,\ldots,x_n)+$$

$$+\sum_{i=1}^{m}\lambda_i\varphi_i(x_1, x_2,\ldots,x_n), \tag{4.27}$$

funcția $\Gamma(X, \Lambda)$ fiind denumită funcția Lagrange, iar scalarii $\lambda_1, \lambda_2,\ldots,\lambda_m$ — multiplicatori Lagrange.

Scanned with OKEN Scanner

Deci, cele două metode transformă problema de optimizare cu restricții într-o problemă fără restricții, a cărei soluționare constă în rezolvarea unui sistem de ecuații normale, obținut prin anularea derivatelor parțiale de ordinul întîi ale funcției obiectiv, care pentru metoda multiplicatorilor Lagrange se scrie:

$$\frac{\partial \Gamma(X,\Lambda)}{\partial x_j}=\frac{\partial F}{\partial x_j}+\sum_{i=1}^{m}\lambda_i\frac{\partial \varphi_i}{\partial x_j}=0 \qquad (j=1,\ldots,n)$$
$$\frac{\partial \Gamma(X,\Lambda)}{\partial \lambda_i}=\varphi_i=0 \qquad (i=1,\ldots,m) \tag{4.28}$$

Sistemul conține $n+m$ ecuații (derivatele parțiale în raport cu λ_i fiind chiar relațiile de restricții), rezultînd cele $n+m$ necunoscute: x_j $(j=1,\ldots,n)$; $\lambda_i(i=1,\ldots,m)$, cel mai accesibil mod de rezolvare a acestuia fiind cel numeric (eliminare Gauss, aproximații succesive Gauss-Seidel, Newton-Raphson etc.).

Relațiile scrise reprezintă condiții necesare, dar nu și suficiente, în cazul general pentru minim. Numai dacă $F(X)$ este o funcție convexă și $\varphi_i(X)$ sînt liniare, se poate arăta că aceste condiții sînt și suficiente pentru minim.

4.2.3.2. Principiul programării liniare

De cele mai multe ori în practică se întîlnesc probleme de optimizare cu restricții. În aceste cazuri, dacă restricțiile și funcția obiectiv sînt funcții liniare de componentele vectorului X și dacă acestor componente li se impun condiții de nenegativitate, modelul de optimizare constituie o problemă de programare liniară.

Dacă funcția obiectiv și una sau toate restricțiile sînt neliniare, atunci problema este de programare neliniară.

Programarea liniară constă în esență în determinarea maximului (minimului) funcției obiectiv, de mai multe variabile necunoscute, legate între ele printr-un sistem cu un număr oarecare de ecuații și inecuații, ce reprezintă restricțiile.

Astfel, se pune problema determinării valorii maxime sau minime a funcției obiectiv:

$$F=c_1x_1+c_2x_2+\ldots+c_nx_n\rightarrow\max\ (\min), \tag{4.29}$$

satisfăcîndu-se restricțiile și condițiile de nenegativitate:

$$\begin{aligned}
&a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n\leqslant b_1\\
&a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n\leqslant b_2\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n\leqslant b_m\\
&x_j\geqslant 0 (j=1,2,\ldots,n)
\end{aligned} \tag{4.30}$$

Sistemul de restricții inegalități (4.30) poate fi transformat într-un sistem de egalități prin atașarea unor *variabile de ecart* $x_{n+i}\geqslant 0$ $(i=1,\ldots m)$ obținîndu-se forma standard a modelului de programare liniară.

Evident că programul liniar standard (cu variabile de ecart) liniar poate fi scris mai concis în forma matriceală:

$$\begin{aligned}
&F=CX\rightarrow\max\ (\min)\\
&AX=B\\
&X\geqslant 0
\end{aligned} \tag{4.31}$$

Scanned with OKEN Scanner

în care: C este vectorul linie al coeficienților funcției obiectiv; B — vectorul coloană al termenilor liberi din sistemul restricțiilor; X — vectorul coloană al variabilelor; A — matricea coeficienților din sistemul restricțiilor.

Mulțimea de $n+m$ valori x_j care satisfac sistemul restricțiilor se numește *soluție*. Dacă în plus aceasta verifică și condițiile de nenegativitate atunci soluția se numește *posibilă*.

Cînd restricțiile sînt liniare independente, iar soluția conține m componente x_j nenule și n componente x_j nule, soluția poartă numele de *soluție de bază*. Dacă această soluție conține un număr de valori x_j nenule mai mic decît m, ea este *degenerată*.

Soluția de bază care face ca funcția obiectiv să fie maximă (minimă) este denumită *soluție optimală*.

Maximizarea funcției obiectiv F se poate transforma în minimizarea funcției:

$$F' = -F = -\sum_{j=1}^{n} c_j x_j \tag{4.32}$$

față de restricțiile date, adică problema minimizării se reduce la o problemă de maximizare prin schimbarea semnului tuturor coeficienților din funcția obiectiv, iar în final prin schimbarea semnului maximului obținut.

Cel mai utilizat algoritm de rezolvare a problemelor de programare liniară este *algoritmul Simplex*, elaborat de Dantzig, care s-a dezvoltat ulterior în mai multe variante.

Principiul acestui algoritm constă în utilizarea în spațiul variabilelor funcției obiectiv a unei figuri geometrice cu un număr de vîrfuri mai mare decît numărul variabilelor, figură denumită *simplex*. Se evaluează funcția obiectiv în fiecare vîrf al acestei figuri și se cercetează apoi direcția ce trece prin vîrful cel mai nefavorabil și centroidul celorlalte vîrfuri. Această cercetare urmărește stabilirea unui nou vîrf favorabil care să-l înlocuiască pe cel mai nefavorabil din simplexul existent. În acest mod se generează un simplex care migrează în cele mai multe cazuri în regiunea extremului funcției obiectiv.

În varianta de bază, algoritmul Simplex presupune o succesiune de iterații în următoarele etape:

E1) Aducerea problemei în forma standard, transformînd inegalitățile în egalități prin introducerea variabilelor de ecart.

E2) Determinarea unei soluții de bază inițiale.

E3) Trecerea la altă soluție de bază, plecînd de la baza inițială sau precedentă și modificînd una din variabile, care nu este în bază, pentru deplasarea la un vîrf ce îmbunătățește funcția obiectiv. În cazul unei probleme de maximizare, se alege variabila cu coeficientul pozitiv cel mai mare, iar în cazul unei probleme de minimizare, se alege variabila cu coeficientul negativ cel mai mare.

E4) Considerarea ca soluție optimală a soluției de bază obținute după etapa anterioară, atunci cînd nu mai este posibil să se micșoreze (mărească) valoarea funcției obiectiv, ceea ce corespunde situației în care toți coeficienții c_j din funcția obiectiv devin pozitivi în varianta minimizării și negativi în cazul maximizării.

Pe baza acestui algoritm s-a elaborat programul ordinator SIMPLEX, cuprinzînd 3 subrutine.

Scanned with OKEN Scanner

Pentru aplicarea algoritmului Simplex se poate utiliza și programul OPALINE, catalogat pe un disc magnetic al sistemelor mari de tip FELIX C și constituit dintr-un modul principal și 8 segmente. Apelarea acestui program se face cu ajutorul unor instrucțiuni de control [29], care preced datele problemei.

Unul din algoritmii utilizați în cazul programării liniare cu variabile întregi este cel al lui Gomory [21].

4.2.3.3. Formularea generală a programării neliniare

Considerații preliminare. *Programarea neliniară* este o metodă matematică de optimizare a problemelor ce pot fi descrise prin expresii neliniare.

În general, pentru rezolvarea problemelor de optimizare condiționată se utilizează programarea neliniară cu restricții, a cărui model matematic este definit de relațiile:

$$F(X)\to \min\ (\max)$$
$$\varphi_j(X)\{\geqslant;\ =\ ;\leqslant\}0 \qquad (j=1,2,\ldots,m) \tag{4.33}$$

unde funcția obiectiv F și funcțiile restricții φ_j constituie împreună funcții continue ale problemei de optimizare, iar X este vectorul variabilelor, de forma:

$$X=\{x_1x_2\ldots x_n\} \tag{4.34}$$

Modelul matematic (4.33) conduce la o problemă de programare neliniară, în următoarele cazuri posibile:

— funcția obiectiv neliniară și restricțiile liniare;
— funcția obiectiv liniară și restricțiile neliniare;
— funcția obiectiv neliniară și restricțiile neliniare.

Înainte de a se descrie cîteva din metodele de soluționare a problemelor de programare neliniară se prezintă condițiile Kuhn-Tucker, obținute prin extinderea metodei clasice a multiplicatorilor Lagrange și pentru cazul restricțiilor de inegalitate.

Condițiile de optimalitate Kuhn-Tucker. Prin generalizarea metodei clasice a multiplicatorilor Lagrange, Kuhn și Tucker au stabilit condițiile care trebuie să fie satisfăcute de un optim cu restricții, local sau global, condiții ce stau la baza celor mai mulți algoritmi existenți pentru rezolvarea problemelor de programare neliniară.

Se consideră problema minimizării funcției:

$$F(X)=F(x_1, x_2,\ldots, x_n) \tag{4.35}$$

în prezența restricțiilor:

$$\varphi_j(X)=\varphi_j(x_1,x_2,\ldots, x_n)\leqslant 0 \qquad (j=1, 2,\ldots, m<n) \tag{4.36}$$

Cu ajutorul a m variabile auxiliare reale w_j, inegalitățile (4.36) se transformă în egalități:

$$\varphi_j(X)+w_j^2=0 \qquad (j=1, 2,\ldots, m) \tag{4.37}$$

Scanned with OKEN Scanner

Aplicînd metoda multiplicatorilor Lagrange se definește funcția auxiliară:

$$\Gamma(X, W, \Lambda)=F(X)+\sum_{j=1}^{m}\lambda_j[\varphi_j(X)+w_j^2] \tag{4.38}$$

care conduce la un punct staționar în ipoteza îndeplinirii condițiilor:

$$\begin{aligned}
&\frac{\partial\Gamma(X, W, \Lambda)}{\partial x_i}=\frac{\partial F(X)}{\partial x_i}+\sum_{j=1}^{m}\lambda_j\frac{\partial\varphi_j(X)}{\partial x_i}=0 \qquad (i=1,2,\ldots, n)\\
&\frac{\partial\Gamma(X, W, \Lambda)}{\partial\lambda_j}=\varphi_j(X)+w_j^2=0 \qquad (j=1, 2,\ldots, m)\\
&\frac{\partial\Gamma(X, W, \Lambda)}{\partial w_j}=2\lambda_j w_j=0 \qquad (j=1, 2,\ldots, m)
\end{aligned} \tag{4.39}$$

Întrucît mărimile w_j sînt variabile auxiliare, acestea pot fi eliminate din sistemul (4.39). Din cea de-a doua relație a acestui sistem rezultă că oricare ar fi j, condiția $\varphi_j(X)=0$ are loc numai dacă $w_j=0$, sau ținînd seama și de ultima relație a aceluiași sistem se obține: $\lambda_j\cdot\varphi_j(X)=0$ $(j=1, 2,\ldots,m)$.

Prin eliminarea variabilelor w_j, rezultă deci condițiile necesare pentru minim:

$$\begin{aligned}
&\frac{\partial F(X)}{\partial x_i}+\sum_{j=1}^{m}\lambda_j\frac{\partial\varphi_j(X)}{\partial x_i}=0 \qquad (i=1, 2,\ldots, n)\\
&\varphi_j(X)\leqslant 0 \qquad\qquad (j=1, 2,\ldots, m)\\
&\lambda_j\cdot\varphi_j(X)=0 \qquad\qquad (j=1, 2,\ldots, m),
\end{aligned} \tag{4.40}$$

cărora Kuhn și Tucker le-au adăugat în plus:

$$\lambda_j\geqslant 0 \qquad\qquad (j=1, 2,\ldots, m), \tag{4.41}$$

relații cunoscute sub numele de *condițiile Kuhn-Tucker*, ce sînt necesare atît pentru un minim local, cît și pentru un minim global.

Dacă funcțiile $F(X)$, $\varphi_j(X)$ și domeniul admisibil sînt convexe, ele devin condiții necesare și suficiente pentru un minim global.

Condițiile de optimalitate (4.40), (4.41) înlocuiesc programul matematic general (4.33) printr-un sistem de ecuații și inecuații a căror expresie matriceală are forma:

$$\begin{aligned}
&\nabla_x(F)+\nabla_x(\Phi^T)=0\\
&\Phi(X)\leqslant 0\\
&\Lambda\Phi(X)=0\\
&\Lambda=0
\end{aligned} \tag{4.42}$$

în care: $F(X)$, $\Phi(X)$ și domeniul admisibil sînt convexe; Λ — vector cu aceleași dimensiuni ca Φ, iar:

$$\nabla_x=\begin{bmatrix}\dfrac{\partial}{\partial x_1}\\ \vdots\\ \dfrac{\partial}{\partial x_n}\end{bmatrix} \tag{4.43}$$

$$\nabla_x(\Phi^T)=\begin{bmatrix}\dfrac{\partial}{\partial x_1}\\ \vdots\\ \dfrac{\partial}{\partial x_n}\end{bmatrix}\{\varphi_1\varphi_2\ldots\varphi_m\} \tag{4.44}$$

Scanned with OKEN Scanner

O problemă de programare convexă admite un domeniu admisibil convex și o funcție obiectiv convexă, lucru asigurat dacă F și φ_j sînt funcții convexe.

Dacă o problemă de programare matematică este cunoscută a fi convexă, prezintă avantajele:

— funcția nu are minime locale;

— condițiile Kuhn-Tucker sînt necesare și suficiente pentru a indica un minim global.

În cele mai multe aplicații concrete din domeniul instalațiilor este dificil de a arăta că funcția analizată este convexă, însă modelele de optimizare obținute în condiții de convexitate pot fi aplicate problemelor de programare matematică neconvexă pentru a găsi un minim local.

Prezentarea generală a metodelor de soluționare a problemelor de programare neliniară. Un algoritm care poate rezolva problema programării neliniare în forma generală (4.33) este de regulă ineficient cînd se utilizează pentru rezolvarea unor probleme cu caracteristici speciale. Din acest motiv, pentru orice problemă este avantajos să se evidențieze toate trăsăturile particulare care ar putea să conducă la o rezolvare mai ușoară.

În [33] Philip Wolfe a făcut o grupare a problemelor de programare după dimensiuni și complexitate, în grupa problemelor neliniare așezînd în ordinea complexității probleme cu variabile separate, probleme de programare pătratică, probleme cu restricții liniare și funcția obiectiv neliniară și probleme cu funcțiile de optimizare explicite și cu derivate ușor de calculat.

Au fost puși la punct numeroși algoritmi, folosiți în anumite cazuri particulare, legate de forma funcțiilor F și φ_j.

Dintre algoritmii de rezolvare a problemelor cu structură specială fac parte: algoritmul lui Wolfe [28], [31] pentru programarea pătratică și algoritmul lui Rosen [31] pentru programarea convexă cu restricții liniare.

Algoritmii de căutare numerică a optimului au la bază:

— metode directe de căutare în domeniul admisibil, ce utilizează evaluări ale funcției obiectiv în punctele din domeniul admisibil al variabilelor de optimizat;

— metode indirecte, prin care problema cu restricții se transformă în una fără restricții, modificînd funcția obiectiv;

— metode de gradient, ce utilizează evaluări atît ale funcției obiectiv, cît și ale gradientului în puncte din domeniul admisibil.

Algoritmii de optimizare parcurg de regulă următoarele trei etape:

E1) Alegerea unui punct de start X_0, punct admisibil în cazul unei probleme cu restricții.

E2) Deplasarea prin puncte succesive X_k ($k=1,2,\ldots$) pînă cînd optimul este atins. De la un punct de bază X_{k-1} la un punct de bază nou X_k se trece prin deplasarea după o direcție $D_k=\{d_1^{(k)}\ d_2^{(k)}\ldots d_n^{(k)}\}$, punctul X_k fiind optimul după această direcție. În cazul metodelor directe, direcțiile de deplasare sînt prevăzute în algoritm, pe cînd la cele de gradient sînt evaluate în fiecare punct X_k.

E3) Verificarea criteriului de convergență, etapă care decide oprirea sau continuarea procesului de căutare a optimului.

Este dificil de a se face afirmații general valabile privind superioritatea unui algoritm față de altul. În consecință, alegerea celui mai bun algoritm, pentru o problemă dată, reprezintă un întreg proces de cercetare.

Scanned with OKEN Scanner

Metoda direcțiilor admisibile. Din categoria metodelor directe privind minimizarea funcțiilor multivariabile în prezența restricțiilor de tip inegalitate, metoda direcțiilor admisibile este cea mai des utilizată, folosind de regulă un procedeu de aproximații succesive de forma:

$$X_{k+1}=X_k+\alpha D_k \tag{4.45}$$

unde direcțiile de deplasare D_k și lungimea pasului de deplasare sînt întotdeauna alese astfel încît X_{k+1} să se afle în interiorul domeniului admisibil (fezabil).

Această metodă conține două etape principale, prima constînd în determinarea pentru fiecare pas k a unui vector D_k care respectă condițiile:

$$\nabla \varphi_j(X)^T D<0; \qquad \nabla F(X)^T D<0, \tag{4.46}$$

iar cea de-a doua determină mărimea pasului α. În funcție de modul în care sînt rezolvate cele două etape, există diverse variante ale metodei [21], [70].

Metoda funcțiilor de penalizare. Metoda funcțiilor de penalizare face parte din categoria metodelor indirecte și reprezintă o tehnică de rezolvare a problemelor de minimizare în prezența relațiilor de restricții, prin transformarea acestora în probleme de minimizare fără restricții.

Minimul neconstrîns al unei probleme de optimizare se poate găsi în interiorul domeniului admisibil sau în exteriorul lui.

a) *Metoda penalizării de punct exterior* utilizează o funcție de penalizare care dă o valoare mare funcției F atunci cînd punctul X iese din domeniul admisibil sau se află în afara lui.

Pentru problema de optimizare neliniară:

$$F(X)\to\min$$
$$\varphi_j(X)\geqslant 0 \quad (j=1, 2,\ldots, m) \tag{4.47}$$

se definește funcția obiectiv de penalizare:

$$\Phi(X, \alpha)=F(X)+\alpha P(X) \tag{4.48}$$

unde α este o constantă, iar funcția de penalizare P este dată de relația generală:

$$P(X)=\begin{cases} 0; & \varphi_j(X)\geqslant 0 \\ \sum\limits_{j=1}^{m}<\varphi_j(X)>^{\beta}; & \varphi_j(X)<0 \end{cases} \tag{4.49}$$

în care β este o constantă nenegativă.

Problema de minimizare neliniară (4.47) s-a transformat într-o problemă de minimizare fără restricții a funcției (4.45).

Prezența termenilor $\alpha\sum\limits_{j=1}^{m}<\varphi_j(X)>^{\beta}$ face ca funcția Φ să ia o valoare mare atunci cînd restricțiile nu mai sînt îndeplinite, altfel spus penalizează încălcarea restricțiilor.

Pentru cazul $0<\beta<1$, funcția Φ este continuă pe frontiera domeniului, putînd avea derivate de primul ordin discontinue pe frontiera domeniului. Dacă $\beta=1$, există un α foarte mare pentru care minimul lui Φ să coincidă cu minimul lui F și din această cauză în practică nu se utilizează $\beta=1$, ci de obicei $\beta=2$.

Scanned with OKEN Scanner

Algoritmul corespunzător acestei metode are următoarele etape:

E1) Se alege un punct inițial X_0 și o valoare inițială α și se determină X_m care minimizează valoarea funcției:

$$\Phi(X, \alpha)=F(X)+\sum_{j=1}^{m}<\varphi_j(X)>^2 \tag{4.50}$$

E2) Se verifică dacă $X_m \in D_a$, unde D_a este domeniul admisibil, prin condiția:

$$|\Delta(X)|=\left(\sum_{i=1}^{n}\delta_i^2\right)^{1/2} \leqslant \varepsilon \tag{4.51}$$

în care:

$$\Delta(X)=X_m(\alpha_{k-1})-X_m(\alpha_k) \tag{4.52}$$

Calculul se oprește dacă este îndeplinită condiția (4.51) sau dacă α este de ordinul lui $10^{-7}...10^{-6}$.

E3) Dacă $X_m \notin D_a$ se alege $\alpha'>\alpha$ și pornind de la $X_0=X_m$ algoritmul reia etapa E1, minimizînd $\Phi(X, \alpha')$.

Pentru un α suficient de mare se forțează minimul funcției să intre în D_a.

b) *Metoda penalizării de punct interior* folosește o funcție obiectiv de forma:

$$\Phi(X, \alpha)=F(X)-\alpha\sum_{j=1}^{m}\frac{1}{\varphi_j(X)} \tag{4.53}$$

în care: F este funcția de optimizat; φ_j — restricțiile care îndeplinesc condițiile $\varphi_j(X) \leqslant 0 \quad (j=1, 2,..., m)$.

Dacă $\alpha>0$, minimul lui $\Phi(X, \alpha)$ se află în domeniul admisibil și converge către minimul lui $F(X)$ pe măsură ce parametrul α se micșorează.

Algoritmul de calcul are următoarele etape:

E1) Se alege punctul inițial X_0 pentru care $\varphi_j(X_0)<0$ și o valoare inițială a lui α.

E2) Se minimizează funcția Φ exprimată prin (4.53) și se determină X_m.

E3) Dacă convergența nu este suficient de bună se micșorează α.

E4) Se alege un nou punct inițial $X_0=X_m$ și se repetă etapa E1.

Metoda gradienților condiționați. Această metodă se utilizează pentru minimizarea unei funcții neliniare într-un domeniu în care minimul unei funcții liniare se obține fără dificultăți.

Fie $F(X)$, $X \in R^n$ o funcție neliniară continuă și derivabilă într-un domeniu convex Ω. Se presupune că gradientul $\nabla F(X)$ a lui $F(X)$ satisface pentru toate punctele din Ω condiția lui Lipschitz:

$$\|\nabla F(X_1)-\nabla F(X_2)\| \leqslant L_0\|X_1-X_2\| \tag{4.54}$$

unde L_0 este constanta Lipschitz.

Algoritmul corespunzător acestei metode cuprinde următoarele etape:

E1) Se alege un punct de start X_0 din domeniul Ω.

E2) Considerînd, în general, aproximația k a procesului iterativ, se minimizează funcția liniară $\nabla F(X_k)^T \cdot Z$ și se determină punctul de minim $Z(X_k)$.

Scanned with OKEN Scanner

E3) Se calculează direcţia de explorare:

$$D_k = Z(X_k) - X_k \tag{4.55}$$

şi

$$\eta(X_k) = \nabla F(X_k)^T D_k \tag{4.56}$$

E4) Se construieşte noul punct de bază:

$$X_{k+1} = X_k + \alpha_k D_k \tag{4.57}$$

unde α_k este lungimea pasului în direcţia D_k egal cu 2^{-i_0}, i_0 fiind primul indice $i=0, 1,\ldots,$ pentru care are loc inegalitatea:

$$F(X_k + 2^{-i} D_k) \leqslant F(X_k) + 2^{-i} \frac{\eta(x_k)}{2} \tag{4.58}$$

E5) Procesul iterativ se opreşte atunci cînd este satisfăcut criteriul:

$$\eta(X_k) = 0, \tag{4.59}$$

iar X_k este soluţia optimă a problemei, care minimizează funcţia $F(X)$.

4.2.4. METODA PROGRAMĂRII DINAMICE

Programarea dinamică reprezintă o metodă de optimizare a sistemelor sau proceselor în care se operează pe faze sau secvenţial, avînd ca aplicare înlocuirea unei probleme de dimensiuni mari printr-o succesiune de probleme de dimensiuni mici, la baza acestora aflîndu-se principiul optimalităţii a lui Bellman [3] care, pe scurt, se formulează astfel: „o politică optimă nu poate fi alcătuită decît numai din subpolitici optime".

Procesele secvenţiale sînt acele procese în cadrul cărora un fluid, un flux de material sau un flux energetic îşi modifică parametrii (temperatură, presiune, umiditate etc.) în cursul unei serii de transformări succesive.

Prin programarea dinamică se pune problema minimizării funcţiei obiectiv:

$$\min Z = \sum_{i=1}^{N} V_i(X_{i-1},\ X_i) \tag{4.60}$$

ştiind că fiecare mărime X_i poate varia într-un domeniu care depinde de X_0 şi X_{i+1}, $\forall i = 1,2,\ldots, N$.

Formula de optimizare (4.60) se generalizează în cazul în care termenii X_i sînt vectori:

$$X_i = \{x_{1i}\ x_{2i}\ \ldots x_{ni}\} \tag{4.61}$$

avînd n componente.

Forma funcţiei Z şi natura domeniilor de variaţie ale variabilelor permit utilizarea unui sistem de N faze pentru care $V_i(X_{i-1},\ X_i)$; $(i=1, 2,\ldots, N)$ este funcţia de valoare ataşată fiecărei faze, iar Z — funcţia de valoare ataşată mulţimii fazelor.

Pentru a găsi valoarea minimă a funcţiei Z se aplică principiul optimalităţii, necesitînd la fiecare pas rezolvarea ecuaţiei funcţionale:

$$f_{0,i}(X_0,\ X_i) = \min_{\{X_{i-1}\}} [V_i(X_{i-1},\ X_i) + f_{0,i-1}(X_0,\ X_{i-1})], \tag{4.62}$$

unde $\{X_{i-1}\}$ înseamnă că X_{i-1} aparţine unei mulţimi de valori ce depinde numai de X_0 şi X_i.

Scanned with OKEN Scanner

Algoritmul de calcul constă în rezolvarea succesivă a problemelor de minimizare (4.62) pentru $i=1,2,\ldots, N$, cu variația discretă a lui X_0 și memorarea rezultatelor intermediare, calculîndu-se astfel subpoliticile optime succesive pentru fazele 1 și 2 împreună, apoi pentru fazele 1, 2 și 3 împreună,..., pentru fazele $\overline{1, N}$ împreună, adică politica sau politicile optime:

$$\min Z = f_{0,N}(X_0, X_N) = \min_{\{X_{n-1}\}} [V_N(X_{N-1}, X_N) + f_{0,N-1}(X_0, X_{N-1})] \qquad (4.63)$$

în care: $f_{0,N}(X_0, X_N)$ este valoarea politicii optime de la X_0 la X_N; $V_N(X_{N-1} X_N)$ — valoarea subpoliticii de la X_{N-1} la X_N;
$f_{0,N-1}(X_0, X_{N-1})$ — valoarea politicii optime de la X_0 la X_{N-1}.

Claritatea ecuațiilor funcționale face ca ele să poată fi aplicate direct, pe cale analitică, fără dificultate.

Cu ajutorul acestei metode se poate da o soluționare optimă alegerii amplasamentului captării și traseului unei magistrale de aducțiune a apei.

4.3. ELEMENTE DE TEORIA GRAFURILOR

4.3.1. NOȚIUNI GENERALE

Ori de cîte ori o structură prezintă o mulțime de obiecte distincte și există o lege care să stabilească o corespondență între acestea, avem de-a face cu un *graf* (fig. 4.7).

Pentru un graf trebuie definite în mod simultan atît mulțimea vîrfurilor $X=\{x_1, x_2,\ldots, X_n\}$, cît și a tranzițiilor Γ, care există între acestea.

Cuplul format din mulțimile X și Γ se numește *graf orientat* și se notează prin $G=(X, \Gamma)$ sau $G=(X, U)$, unde U este mulțimea *arcelor*, reprezentate prin săgeți.

Tranzițiile de forma $(x_i \rightarrow x_j)$ au originea în vîrful x_i și extremitatea în vîrful x_j și se mai notează sub formele: (x_i, x_j), u_j^i sau $x_i \in \Gamma(x_j)$.

Grafurile orientate permit tranziții de un singur sens. În acest caz, între două vîrfuri consecutive este un singur arc cu săgeata într-un anumit sens.

Arcele care au un vîrf comun se numesc arce adiacente, iar extremitățile unui arc se numesc vîrfuri adiacente.

Un arc este incident la un vîrf dacă vîrful respectiv este o extremitate a arcului.

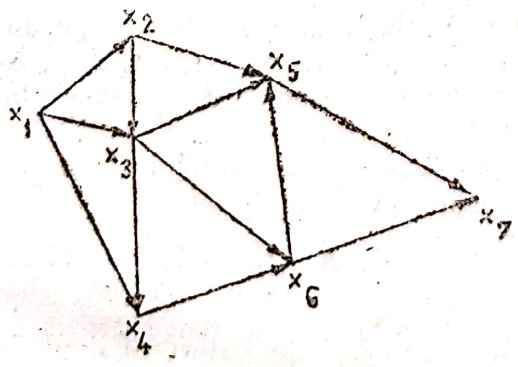


Fig. 4.7. Graf orientat

O succesiune de arce adiacente formează un *drum*, care dacă trece o singură dată prin vîrfurile sale se numește drum elementar, dacă se întoarce în punctul de plecare se numește circuit, iar dacă trece o dată și numai o dată prin fiecare vîrf al grafului se numește drum hamiltonian.

Există diferite metode pentru căutarea drumurilor și circuitelor hamiltoniene (metoda Kaufmann, teorema König, teorema Dirac).

Scanned with OKEN Scanner

Lungimea drumului este dată, prin convenție, de numărul arcelor sale. Pentru calculul lungimii minime a drumului de la un vîrf x_i la toate celelalte vîrfuri ale unui graf G se folosește algoritmul Dantzig [106].

Graful orientat care nu are circuite este denumit *arborescență*.

Numărul ciclomatic al unui graf G dă numărul maxim de cicluri independente c și se scrie analitic cu ajutorul relației lui Euler:

$$c=a-(v-1) \tag{4.64}$$

în care: a este numărul de arce; v — numărul total de vîrfuri.

La un *graf conjugat*, un număr egal de arce se obține înlocuind fiecare vîrf cu un circuit și fiecare circuit cu un vîrf.

Numărul ciclomatic independente c_* ale grafului conjugat G_* este dat de relația:

$$c_*=a-c=v-1 \tag{4.65}$$

Operația prin care fiecărui arc i se atribuie o valoare poartă numele de valorizarea arcelor.

Valoarea unui drum este egală cu suma valorilor arcelor ce formează drumul. Prin valorizarea arcelor unui graf se poate determina drumul de valoare optimă din graf folosind diverși algoritmi: Ford, Bellman-Kalaba, Yen.

Graful neorientat nu presupune relații univoce între elementele sale, iar vîrfurile sînt legate prin *muchii*, care se reprezintă fără săgeți. În locul drumului apare *lanțul*, a cărui lungime este dată de numărul muchiilor sale.

Într-un graf neorientat gradul unui vîrf x_j este numărul muchiilor care se leagă de vîrful x_j.

Pentru un graf neorientat cu m muchii, suma gradelor vîrfurilor sale este egală cu $2m$.

Un graf complet are proprietatea că oricare două vîrfuri sînt unite printr-o muchie, în cazul grafurilor neorientate și că oricare două vîrfuri ale sale x_i și x_j sînt unite printr-un arc (x_i, x_j) sau printr-un arc (x_j, x_i) ori prin ambele arce (x_i, x_j) și (x_j, x_i), în cazul grafurilor orientate.

Un graf se numește *conex* dacă pentru oricare două vîrfuri ale sale există în graf cel puțin un drum (lanț) care le leagă, iar în caz contrar graful este *neconex*.

Graful neorientat conex fără cicluri se numește *arbore*. Muchiile complementare muchiilor grafului definesc *coarborele*.

Dacă $G=(X, U)$ este un arbore cu v vîrfuri și m muchii, atunci:

$$m=v-1 \tag{4.66}$$

Un graf $G=(X, \Gamma)$ poate conține un graf parțial care să fie și arbore (numit arbore parțial) numai dacă este conex. Graful parțial al lui $G=(X, \Gamma)$ este prin definiție un graf $\bar{G}=(X, \Delta)$ în care $\forall x_i$, $\Delta x_i \subset \Gamma x_i$.

Scanned with OKEN Scanner

4.3.2. ALGORITMI PENTRU DETERMINAREA UNUI ARBORE DE VALOARE MINIMĂ

4.3.2.1. Algoritmul Sollin

Se consideră graful conex $G=(X, U)$ căruia i se atașează matricea valorilor M ale cărei elemente m_{ij} sînt:

$$m_{ij}=\begin{cases}\lambda(u_j^i), \text{ dacă vîrfurile } x_i \text{ și } x_j \text{ sînt adiacente}\\ \infty, \text{ dacă vîrfurile } x_i \text{ și } x_j \text{ nu sînt adiacente}\\ 0, \text{ dacă } i=j\end{cases} \tag{4.67}$$

unde $\lambda(u_j^i)$ este valoarea atribuită muchiei $u_j^i \in U$.

Algoritmul Sollin pentru determinarea arborelui de valoare minimă presupune parcurgerea următoarelor etape:

E1) Se parcurg într-o ordine oarecare vîrfurile grafului, constituindu-se arbori ai subgrafelor grafului dat, formate din vîrful considerat și vîrful legat de el prin muchia de valoare minimă, arborii rezultați fiind neconectați între ei.

E2) Arborii rezultați se consideră a fi vîrfuri și se revine la etapa E1.

E3) Procesul de calcul se consideră terminat în momentul cînd se obține un singur arbore, corespunzător unui singur subgraf, care este însuși graful dat.

Acest algoritm este o metodă aproximativă, conducînd în general la o soluție apropiată de cea optimă.

4.3.2.2. Algoritmul Kruskal

Algoritmul Kruskal este o metodă exactă de determinare a arborelui de valoare minimă într-un graf, necesitînd următoarele etape principale:

E1) Se alege o muchie a grafului cu valoarea cea mai mică.

E2) Dintre muchiile nealese, se alege una care nu formează circuite cu muchiile deja alese și care are valoarea cea mai mică.

E3) Procesul de calcul se consideră terminat în momentul în care s-a determinat mulțimea $\bar{U}$ formată din cele v-1 muchii ale arborelui:

$$\bar{U}=\{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{v-1}\}, \tag{4.68}$$

iar graful $\bar{G}=(X, \bar{U})$ este un arbore de valoare minimă.

Dacă toate valorile $\lambda(u_j^i)$ pentru $u_j^i \in U$ sînt distincte, există o soluție a problemei și numai una. În ipoteza existenței unor valori egale la mai multe muchii, problema poate avea mai multe soluții.

Acest algoritm a fost implementat în cadrul programului ordinator *ARBMIN*, putînd fi utilizat pentru determinarea configurației optime a rețelelor ramificate de distribuție a energiei electrice, termice, a gazelor și a apei.

4.3.3. ALGORITMI PENTRU DETERMINAREA UNUI DRUM DE VALOARE MINIMĂ

Principalii algoritmi de determinare a unui drum de valoare minimă din graf au la bază principiul optimalității, potrivit căruia orice drum optim în graf este format din subdrumuri optime din acel graf. Acești algoritmi se utilizează pentru determinarea traseului optim al unor magistrale de aducțiune a apei și a gazelor.

Scanned with OKEN Scanner

4.3.3.1. Algoritmul Ford

Se consideră un graf conex orientat $G=(X, U)$ și fiecărui arc $u_j^i \in U$ i se asociază un număr $\lambda(u_j^i) \geqslant 0$ reprezentînd valoarea arcului. Grafului G i se atașează matricea M, ale cărei elemente m_{ij} sînt date de relațiile (4.67).

Drumul μ din graful de valoare minimă are valoarea totală:

$$\lambda(\mu)=\sum_{u_j^i \in \mu} \lambda(u_j^i) \to \min \tag{4.69}$$

și se află, conform algoritmului Ford, în următoarele etape:

E1) Se numerotează vîrfurile grafului într-o ordine oarecare: $x_0, x_1, \ldots x_n$ în așa fel încît vîrful origine să fie x_0, iar vîrful extremitate să fie x_n.

E2) Se atribuie fiecărui vîrf $x_i (i=\overline{0, n})$ o marcă l_i, pe scurt $x_i(l_i)$ care să reprezinte lungimea unui drum arbitrar de la x_0 la x_i; inițial se ia $l_0=0$ și $l_i=\infty$, dacă $i \neq 0$.

E3) Se consideră pe rînd toate arcele (x_i, x_j) din U, se calculează diferența $l_j - l_i$ și se compară cu $\lambda(x_i, x_j)$, iar dacă:

$$l_j - l_i > \lambda(x_i, x_j) \tag{4.70}$$

atunci se înlocuiește de fiecare dată l_j prin:

$$l_j' = l_i + \lambda(x_i, x_j) < l_j \tag{4.71}$$

obținîndu-se o reducere pentru marca vîrfului x_j.

E4) Se definește pentru uniformitate marcajul:

$$l_j' = \begin{cases} l_j, \text{ dacă } l_j - l_i \leqslant \lambda(x_i, x_j) \\ l_i + \lambda(x_i, x_j), \text{ dacă } l_j - l_i > \lambda(x_i, x_j) \end{cases} \tag{4.72}$$

E5) Se continuă această reducere pînă cînd pentru orice arc din U avem $l_j - l_i \leqslant \lambda(x_i, x_j)$.

E6) Se determină vîrfurile prin care trece drumul de valoare minimă: se găsește vîrful $x_{p_1}(l_n)$ cu proprietatea $l_n - l_{p_1} = \lambda(x_{p_1}, x_n)$, deci arcul (x_{p_1}, x_n) este ultimul arc din drum, apoi se găsește vîrful $x_{p_2}(l_{p_1})$ cu proprietatea $l_{p_1} - l_{p_2} = \lambda(x_{p_2}, x_{p_1})$, deci arcul (x_{p_2}, x_{p_1}) este penultimul arc din drum, ș.a.m.d. pînă se ajunge la vîrful x_0.

Drumul cu valoarea minimă l_n este:

$$\mu=(x_0, x_{pk}, x_{pk-1}, \ldots, x_{p_1}, x_n) \tag{4.73}$$

Schema logică a acestui algoritm este prezentată în figura 4.8, unde: N este ordinul grafului; M_1 — număr foarte mare; $M(I\ J)$ — matricea grafului; $L(I)$ — vector ale cărui componente dau mărcile vîrfurilor.

4.3.3.2. Algoritmul Bellman-Kalaba

O variantă a algoritmului precedent, prezentată cu ajutorul programării dinamice, este algoritmul Bellman-Kalaba.

Dacă se notează cu V_i valoarea minimă a drumurilor $\mu_n^i (i=\overline{1, n})$ existente de la vîrful x_i la x_n:

$$V_i = \lambda(\mu_n^i),\ (i=\overline{1, n}) \tag{4.74} \qquad V_n = 0, \tag{4.75}$$

Scanned with OKEN Scanner

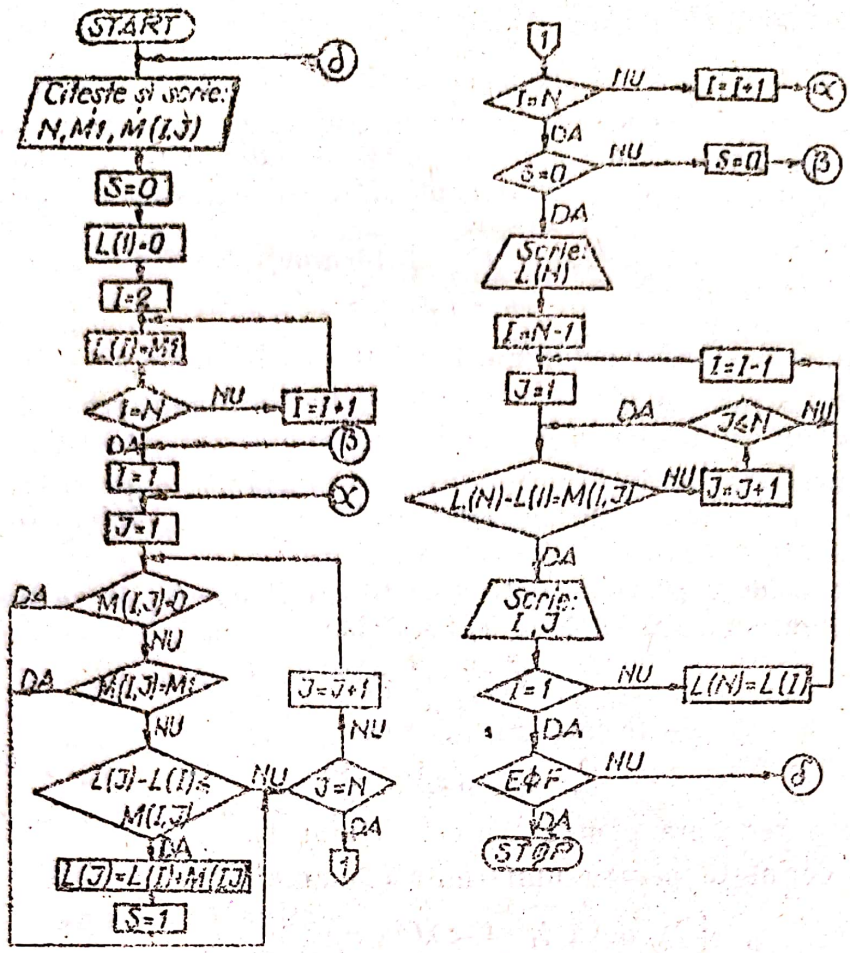


Fig. 4.8. Schema logică a algoritmului Ford de determinare a drumului de valoare minimă într-un graf

atunci conform principiului optimalității:

$$V_i=\min_{j\neq i}(V_j+m_{ij}),\ (i=\overline{1,\ n-1};\ j=\overline{1,\ n});\ V_n=0 \tag{4.76}$$

Pentru rezolvarea sistemului (4.76) se procedează iterativ, notînd cu V_i^k valoarea lui V_i obținută la iterația k, și anume:

$$V_i^0=m_{in},\ (i=\overline{1,\ n-1});\quad V_n^0=0 \tag{4.77}$$

Se calculează:

$$V_i^1=\min_{j\neq i}(V_j^0+m_{ij}),\ (i=\overline{1,\ n-1});\ j=\overline{1,\ n};\quad V_n^1=0 \tag{4.78}$$

și apoi:

$$V_i^k=\min_{j\neq i}(V_j^{k-1}+m_{ij}),\ (i=\overline{1,\ n-1};\quad j=\overline{1,\ n});\ V_n^k=0 \tag{4.79}$$

Iterația de ordinul k exprimată prin relațiile (4.79) dă valori finite doar pentru drumurile de lungime cel mult $k+1$ care ajung la x_n, alegîndu-se dintre acestea cele minimale.

De la o iterație la următoarea:

$$V_i^k\leqslant V_i^{k-1},\quad \forall j \tag{4.80}$$

Scanned with OKEN Scanner

Numerele V_i^k ($i \neq n$, $k=0, 1, \ldots$) formează șiruri monoton descrescătoare, încît se ajunge în mod necesar la un minim, după un număr finit de iterații care nu depășește $n-1$. Deci, algoritmul se oprește cînd se ajunge la o iterație k, astfel încît $V_i^k = V_i^{k+1}$ ($i=\overline{1, n}$), iar valoarea minimă a drumului între vîrfurile x_0 și x_n este $V_1^k = V_1^{k+1}$.

Pentru identificarea drumurilor care au valorile minime găsite, se deduce din (4.79) că de-a lungul lor, la ultima iterație:

$$V_i^k = m_{ij} + V_j^{k-1} = m_{ij} + V_j^k \tag{4.81}$$

În figura 4.9 se prezintă schema logică a programului ordinator DRUMIN, elaborat pe baza acestui algoritm, unde: N este ordinul grafului; $M(I, J)$ — matricea asociată grafului; $V(I, K)$ — vectorul coloană construit la fiecare iterație K; $X(I)$ — succesiunea vîrfurilor drumului de valoare minimă; VAL — valoarea drumului minim în graf.

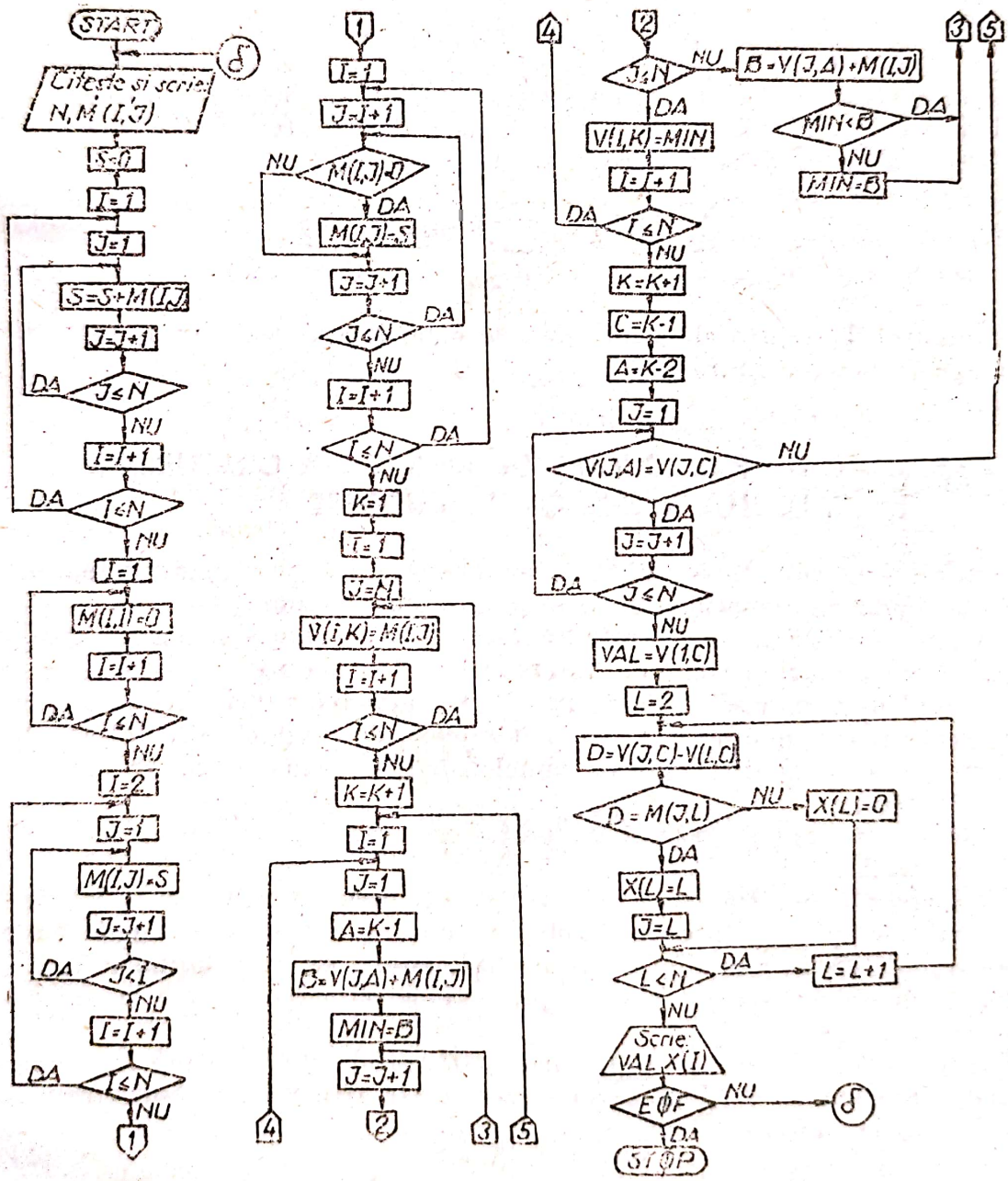


Fig. 4.9. Schema logică a algoritmului Bellman-Kalaba de determinare a drumului de valoare minimă într-un graf

Scanned with OKEN Scanner

4.3.4. REȚEAUA DE TRANSPORT

Rețeaua de transport reprezintă un graf finit fără bucle, în care fiecărui arc u îi este asociat un număr $c(u)>0$ numit capacitatea arcului și în care:

— există un vîrf x_0 (intrarea în rețea) și numai unul, astfel încît $\Gamma^{-1}x_0=\varnothing$;

— există un vîrf x_n (ieșirea din rețea) și numai unul, astfel încît $\Gamma x_n=\varnothing$;

Fie $U_{x_i}^-$ mulțimea arcelor incidente interior în x_i și $U_{x_i}^+$ aceea a arcelor incidente în exterior în x_i.

O funcție în valori întregi $\Phi(u)$ definită pe U este un *flux* pentru o rețea de transport dacă sînt satisfăcute relațiile:

$$\Phi(u)\geqslant 0 \qquad \forall u\in U \tag{4.82}$$

$$\sum_{u\in U_{x_i}^-}\Phi(u)=\sum_{u\in U_{x_i}^+}\Phi(u) \qquad (x_i\neq x_0, \quad x_i\neq x_n) \tag{4.83}$$

$$\Phi(u)\leqslant c(u) \qquad \forall u\in U \tag{4.84}$$

Funcția $\Phi(u)$ poate fi asimilată cu o cantitate de materie care trece prin arcul u și care aici nu trebuie să depășească capacitatea $c(u)$ a acestui arc.

Din relația (4.83) se deduce:

$$\sum_{u\in U_{x_0}^+}\Phi(u)=\sum_{u\in U_{x_0}^-}\Phi(u)=\Phi_0 \tag{4.85}$$

Numărul Φ_0 reprezintă cantitatea de materie care iese din x_0 spre rețea și ajunge prin rețea în x_n.

4.3.5. CONDIȚII PARTICULARE DE APLICARE A GRAFURILOR LA CALCULUL REȚELELOR DE CURGERE

Orice *rețea de curgere*, (hidraulică, termică, electrică) poate fi reprezentată, ca model matematic, printr-un graf topologic conex.

Graful topologic al rețelei este reprezentarea în plan a structurii spațiale alcătuite din noduri și conducte interconectate în noduri.

Gradul unui nod este dat de numărul conductelor incidente în acel nod și nu poate fi mai mic decît 1, întrucît rețelele de conducte sînt întotdeauna grafuri conexe. Gradul 0 îl are un punct izolat, iar nodurile cu gradul 1 sînt noduri periferice.

Rețeaua inelară reprezintă un graf conex care nu are vîrfuri cu gradul mai mic decît 2.

Folosirea și analiza relațiilor dintre elementele rețelelor de conducte ca și grafuri au aplicabilitate în rezolvarea unor probleme ca alegerea formei rețelelor, aprecierea gradului de siguranță, modul de transformare legat de metodele de calcul, determinarea regimului de curgere în rețea și probleme economico-energetice.

În conformitate cu teoria grafurilor, în graful rețelelor, numărul maxim de inele independente M în fiecare sistem este egal cu numărul ciclomatic dat de relația (4.64):

$$M=T-N+1 \tag{4.86}$$

în care: T este numărul tronsoanelor; N — numărul nodurilor rețelei.

Scanned with OKEN Scanner

În dezvoltarea modernă a studiilor de specialitate, grafurile se utilizează ca modele matematice pentru rezolvarea problemelor tehnice și economico-energetice ale rețelelor de alimentare cu apă, cu gaze, cu energie termică și electrică prin calcul matriceal.

Metoda teoriei grafurilor permite să se efectueze înregistrarea tuturor proprietăților geometrice principale și a formei rețelei prin întocmirea unor matrice.

Astfel, graful neorientat al unei rețele poate fi descris prin *matricea tranzițiilor* $[A_t]_{N,N}$ având numărul de linii și de coloane egal cu numărul de vîrfuri N și elementele a_{ij} date de relațiile:

$$a_{ij}=\begin{cases} 1, \text{ dacă vîrfurile } i \text{ și } j \text{ sînt adiacente} \\ 0, \text{ dacă vîrfurile } i \text{ și } j \text{ nu sînt adiacente} \end{cases} \tag{4.87}$$

Neexistînd bucle, elementele diagonalei principale sînt nule. Această matrice permite determinarea gradului nodurilor, adică numărul de conducte convergente în nod, prin însumarea elementelor nenule pe linii sau pe coloane.

Matricea de incidență (matricea vîrfurilor) $[A_v]_{N,T}$ stabilește legătura între arcele și vîrfurile grafului orientat al unei rețele, ilustrînd interconectarea nodurilor în rețea și avînd pe linii numărul de ordine al vîrfurilor N, iar pe coloane cel al arcelor T. Elementele a_{ij} ale acestei matrice sînt definite de relațiile:

$$a_{ij}=\begin{cases} 1 \text{ dacă arcul } j \text{ are incidență în vîrful } i \\ 0, \text{ dacă arcul } j \text{ nu are incidență în vîrful } i \end{cases} \tag{4.88}$$

Matricea de apartenență (matricea ciclurilor) $[A_c]_{M,T}$ stabilește legătura între arcele și ciclurile grafului orientat al unei rețele, indicînd apartenența conductelor la inelele rețelei și avînd pe linii numărul de ordine al inelelor M, iar pe coloane cel al arcelor T. Elementele a_{ij} ale acestei matrice sînt definite prin relațiile:

$$a_{ij}=\begin{cases} 1, \text{ dacă arcul } j \text{ aparține ciclului } i \\ 0, \text{ dacă arcul } j \text{ nu aparține ciclului } i \end{cases} \tag{4.89}$$

Topologia unei rețele de conducte poate fi descrisă complet și univoc cu ajutorul matricei de incidență și a matricei de apartenență.

Avînd în vedere că majoritatea elementelor matricelor atașate grafului, definite sub formele menționate, sînt nule și că acestea ocupă un volum extrem de mare în memoria calculatoarelor electronice, aplicarea în calcul automatizat al rețelelor a acestor matrice se poate face sub o formă particularizată.

Dintre elementele matricei A_v se reține numai elementele nenule, precizîndu-se pentru fiecare arc j nodul inițial NI_j și nodul final NF_j, spre care are loc curgerea. Pe lîngă faptul că se realizează astfel și prezentarea elementelor matricei A_t, eliminîndu-se perechile de noduri care nu sînt interconectate prin conducte, există posibilitatea de orientare a mișcării fluidului în conducte

Scanned with OKEN Scanner

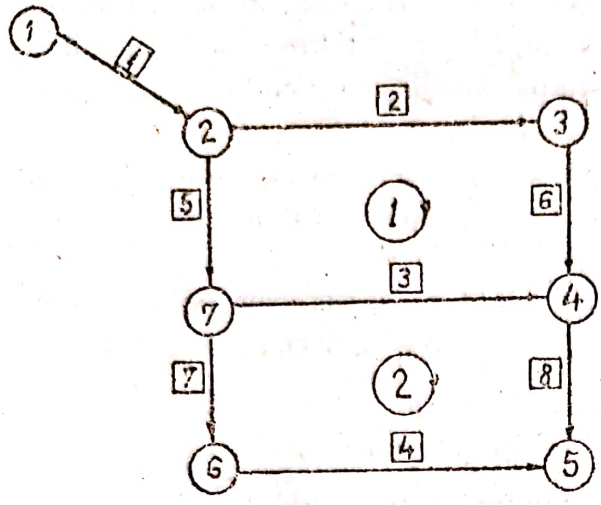


Fig. 4.10. Graful topologic al unei rețele inelare de conducte

Dintre elementele matricei A_c se reţin, de asemenea, numai elementele nenule, prin nominalizarea tuturor nodurilor care formează un inel. Pentru calculul orientării se consideră întotdeauna acelaşi sens de parcurgere, de exemplu sensul orar, începînd cu un nod arbitrar şi încheind ciclul cu acelaşi nod.

Pentru exemplul de reţea cu graful topologic din figura 4.10, elementele nenule ale matricelor A_v şi A_t asociate grafului, care se reţin, se scriu astfel:

j	1	2	3	4	5	6	7	8
NI_j	1	2	7	6	2	3	7	4
NF_j	2	3	4	5	7	4	6	5

iar matricea A_c are forma particulară:

$$A_c = \begin{bmatrix} 2 & 3 & 4 & 7 & 2 \\ 7 & 4 & 5 & 6 & 7 \end{bmatrix}$$

adică se înscriu 26 elemente nenule, în loc să se înscrie în total 121 elemente, din care 49 ale matricei A_t, 56 ale matricei A_v şi 16 ale matricei A_c.

Scanned with OKEN Scanner

5

Modele de optimizare pentru calculul instalațiilor

5.1. CALCULUL HIDRAULIC OPTIM AL CANALELOR CU TALUZURI PLANE ȘI RADIER CILINDRIC

5.1.1. EXPRIMAREA ELEMENTELOR GEOMETRICE ALE SECȚIUNII TRANSVERSALE

Ținînd seama de neajunsurile legate de construirea canalelor cu secțiunea trapezoidală, de necesitatea industrializării sistemelor, precum și de avantajele din punct de vedere hidraulic, în practica inginerească s-a adoptat, pe lîngă alte forme de profile ale canalelor deschise (parabolice, semieliptice), profilul cu taluzuri plane și radier cilindric.

Cu notațiile din figura 5.1 și condiția de tangență a laturilor oblice ale trapezului la arcul de cerc, elementele secțiunii transversale au expresiile:

$$m=\operatorname{ctg}\theta \tag{5.1}$$

$$b=2\ r\ \sin\theta=f_1(m)r \tag{5.2}$$

$$h=2\ r\ \sin^2(\theta/2)=f_2(m)r \tag{5.3}$$

$$H-h=\alpha r \tag{5.4}$$

$$H=\frac{\alpha+f_2(m)}{f_2(m)}\ h=[\alpha+f_2(m)]r \tag{5.5}$$

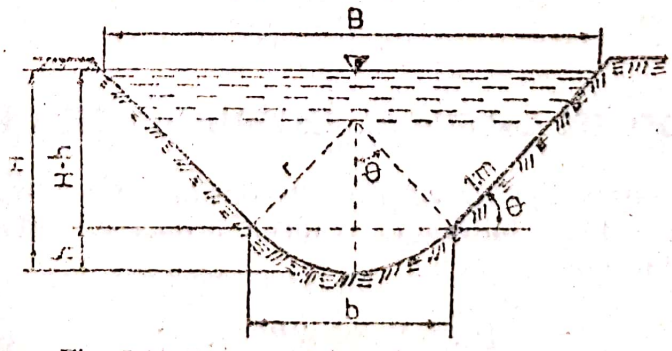


Fig. 5.1. Elementele secțiunii transversale

Scanned with OKEN Scanner

$$B=b+2m(H-h)=[2m\alpha+f_1(m)]r \tag{5.6}$$

$$f_1(m)=\frac{4}{m'} \tag{5.7}$$

$$m'=2\sqrt{1+m^2} \tag{5.8}$$

$$f_2(m)=1-\frac{2m}{m'} \tag{5.9}$$

$$\alpha=\frac{H}{r}-f_2(m)=f_2(m)\left(\frac{H}{h}-1\right) \tag{5.10}$$

Ţinînd seama de relaţiile (5.5) şi (5.6) se obţine:

$$\beta=\frac{B}{H}=\frac{2m\alpha+f_1(m)}{\alpha+f_2(m)} \tag{5.11}$$

$$\alpha=\frac{f_1(m)-f_2(m)\beta}{\beta-2m} \tag{5.12}$$

Potrivit relaţiei (5.10), cînd $h=H$ (profil circular) şi $h=0$ (profil trapezoidal), α trece de la 0 la ∞.

Introducînd în relaţia (5.11) condiţiile $\alpha=0$ şi $\alpha=\infty$, se obţine condiţia de compatibilitate a problemei:

$$2m<\beta<\frac{4}{m'-2m} \tag{5.13}$$

ce exprimă domeniul în care trebuie analizată legătura dintre elementele geometrice şi hidraulice.

Expresiile perimetrului udat P_u, ariei A şi razei hidraulice R a secţiunii transversale, rezultă:

$$P_u=2\sqrt{1+m^2}(H-h)+2\theta r=[m'\alpha+2f_3(m)]r \tag{5.14}$$

$$A=[b+m(H-h)](H-h)+\frac{2\theta-\sin 2\theta}{2}r^2=[m\alpha^2+f_1(m)\alpha+f_4(m)]r^2 \tag{5.15}$$

$$R=\frac{A}{P_u}=\frac{m\alpha^2+f_1(m)\alpha+f_4(m)}{m'\alpha+2f_3(m)}r \tag{5.16}$$

în care:

$$f_3(m)=\operatorname{arctg}\frac{1}{m} \tag{5.17}$$

$$f_4(m)=f_3(m)-\frac{2m}{m'} \tag{5.18}$$

Introducînd notaţiile:

$$\varphi=\alpha+f_2(m) \tag{5.19}$$

$$\tau=2m\alpha+f_1(m) \tag{5.20}$$

$$\psi=\sqrt{m\alpha^2+f_1(m)\alpha+f_4(m)} \tag{5.21}$$

$$f=m'\alpha+2f_3(m) \tag{5.22}$$

fiecare mărime geometrică dimensională se poate exprima în funcţie de numai o singură mărime dimensională şi alte două mărimi adimensionale m şi α, conform tabelului 5.1.

5.1.2. MODEL DE CALCUL AL PROFILULUI OPTIM HIDRAULIC

Exprimînd viteza apei V cu ajutorul formulei Chézy şi adoptînd pentru coeficientul de rezistenţă hidraulică C formula lui Pavlovski, rezultă expresia cunoscută a debitului:

$$Q=AV=\frac{1}{n}AR^{y+0,5}i^{0,5} \tag{5.23}$$

Scanned with OKEN Scanner

Tabelul 5.1

Expresia mărimilor geometrice principale

Mărimea	în funcție de:					
	r	H	B	A	P_u	R
0	1	2	3	4	5	6
r	r	$\frac{1}{\varphi}H$	$\frac{1}{\tau}B$	$\frac{1}{\psi^2}\sqrt{A}$	$\frac{1}{f}P_u$	$\frac{f}{\psi^2}R$
H	φr	H	$\frac{\varphi}{\tau}B$	$\frac{\varphi}{\psi}\sqrt{A}$	$\frac{\varphi}{f}P_u$	$\frac{f\varphi}{\psi^2}R$
B	τr	$\frac{\tau}{\varphi}H$	B	$\frac{\tau}{\psi}\sqrt{A}$	$\frac{\tau}{f}P_u$	$\frac{f\tau}{\psi^2}R$
A	$\psi^2 r$	$\frac{\psi^2}{\varphi^2}H^2$	$\frac{\psi^2}{\tau^2}B^2$	A	$\frac{\psi^2}{f^2}P_u$	$\frac{f^2}{\psi^2}R^2$
P_u	fr	$\frac{\varphi}{f}H$	$\frac{f}{\tau}B$	$\frac{f}{\psi}\sqrt{A}$	P_u	$\frac{f^2}{\psi^2}R$
R	$\frac{\psi^2}{f}r$	$\frac{\psi^2}{f\varphi}H$	$\frac{\psi^2}{f\tau}B$	$\frac{\psi}{f}\sqrt{A}$	$\frac{\psi^2}{f^2}P_u$	R

în care: n este coeficientul de rugozitate; i — panta longitudinală; y — exponent în formula Pavlovski, dat de relația:

$$y=2{,}5\sqrt{n}-0{,}13-0{,}75\sqrt{R}(\sqrt{n}-0{,}10) \tag{5.24}$$

Introducînd în (5.23) expresia mărimilor A și R funcție de H din tabelul 5.1, se obține relația de bază pentru calculul hidraulic al canalelor cu taluzuri plane și radier cilindric în regim permanent și uniform, sub forma:

$$Q=\frac{1}{n}\frac{\psi^{y+2,5}}{f^{y+0,5}\varphi^{y+2,5}}H^{y+2,5}i^{0,5} \tag{5.25}$$

Dacă în relația (5.14) se înlocuiește expresia lui r obținută din relația (5.15) și se face notația:

$$\lambda=\frac{f}{\psi}=\frac{m'\alpha+2f_3(m)}{\sqrt{m\alpha^2+f_1(m)\alpha+f_4(m)}} \tag{5.26}$$

expresia (5.14) a perimetrului udat primește forma:

$$P_u=\lambda\sqrt{A} \tag{5.27}$$

În cazul profilului optim hidraulic, care corespunde perimetrului udat minim la aceleași valori A, n și i, parametrul adimensional λ devine minim, iar $\alpha=\alpha_0$ se obține prin determinarea minimului funcției (5.26) din condiția $\partial\lambda/\partial\alpha=0$, respectiv,:

$$\alpha_0=\frac{f_1(m)f_3(m)-m'f_4(m)}{\frac{1}{2}m'f_1(m)-2mf_3(m)}=\frac{2m}{m'}=\frac{m}{\sqrt{1+m^2}} \tag{5.28}$$

Scanned with OKEN Scanner

Introducînd relația (5.28) în relația (5.11) rezultă condiția profilului optim hidraulic de taluz dat:

$$\beta_0 = m' = 2\sqrt{1+m^2} \tag{5.29}$$

Dacă în urma calculelor de dimensionare, viteza apei nu se încadrează între limitele admisibile V_{min} și V_{max} se mărește sau se micșorează panta i, conform relației (5.30), iar adîncimea apei H se determină cu relația (5.31):

$$i = \frac{n^2 V^2_{min\,(max)}}{R^{2y+1}} \tag{5.30}$$

$$H = \frac{\varphi}{\psi}\sqrt{\frac{Q}{V_{min\,(max)}}} \tag{5.31}$$

Schema logică a programului ordinator OPTIM 1, elaborat pentru rezolvarea problemelor practice de calcul al profilului optim hidraulic de tipul menționat, se prezintă în figura 5.2, unde ε este eroarea admisă, iar identificatorului KOD i se atribuie valoarea 1 cînd se cunoaște $Q(W=Q$ și $T=0)$, respectiv valoarea 0, cînd se cunoaște $H(W=H$ și $T=0)$ sau se cunoaște $B(W=0$ și $T=B)$.

5.2. OPTIMIZAREA PROIECTĂRII REȚELELOR EXTERIOARE DE DISTRIBUȚIE A APEI

5.2.1. ALGORITM DE OPTIMIZARE A TRASEULUI REȚELELOR RAMIFICATE

În cadrul proceselor de optimizare ce le necesită proiectarea rațională a rețelelor ramificate de distribuție a apei, care să conducă la folosirea judicioasă a fondurilor de investiții, la o exploatare normală cu consum minim de energie și cheltuieli reduse, de o mare importanță este optimizarea traseului acestora.

Modelarea acestei probleme se realizează prin reprezentarea grafului conex neorientat, alcătuit din instalația de punere sub presiune ca origine, nodurile de consum ca vîrfuri și traseele arterelor ce unesc aceste puncte ca muchii. În mod similar se pune problema și pentru rețelele termice, electrice și de gaze.

Fie graful conex neorientat $G=(X, U)$ cu $X=\{1, 2, \ldots, n\}$, în care fiecare muchie $u^i_j \in U$ are asociată o valoare $\lambda(u^i_j)>0$, în unități convenționale, după un criteriu de optimizare adoptat. Acestui graf i se atașează matricea C de ordin n ale cărei elemente sînt:

$$c_{ij} = \begin{cases} \lambda(u^i_j), & \text{dacă } u^i_j \in U \\ \infty, & \text{dacă } u^i_j \notin U, \text{ sau } i=j. \end{cases} \tag{5.32}$$

Indicii vîrfurile muchiilor u^i_j ce aparțin grafului parțial al arborilor minimi și valorile corespunzătoare c_{ij} se reține într-o matrice M, care potrivit faptului că un arbore minim are n-1 muchii, va avea 3 coloane și cel puțin n-1 linii. Această matrice se construiește în următoarele etape:

E1) Se determină elementele minime de pe fiecare linie a matricei C.

E2) Se alege una din liniile pentru care elementul minim este unic și se notează cu r, s indicii săi.

Scanned with OKEN Scanner

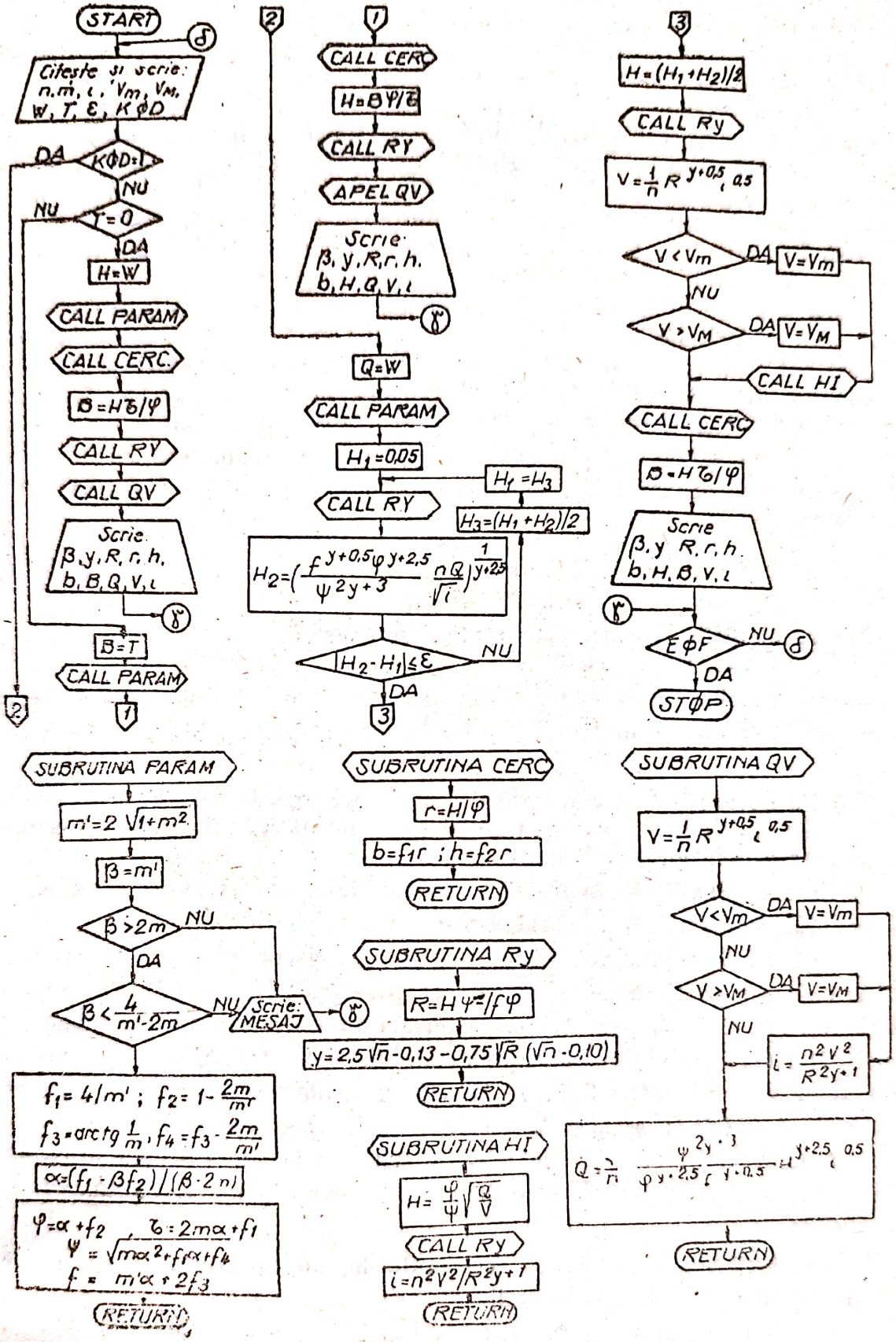


Fig. 5.2. Schema logică a algoritmului de calcul al profilului optim hidraulic

Scanned with OKEN Scanner

E3) Se înregistrează pe prima linie a matricei M valorile r, s, c_{rs} și se marchează elementele c_{rs}, c_{sr} din matricea C.

E4) Se determină minimul elementelor nemarcate de pe liniile matricei C pe care există cel puțin un element marcat și se notează cu r, s indicii elementului minim sau a unuia dintre ele în cazul că există mai multe.

E5) Dacă pe linia s a matricei C există elemente marcate, se marchează și elementele c_{rs}, c_{sr} și se trece la etapa E6, iar dacă pe linia s a matricei C nu există elemente marcate, pentru fiecare element c_{ts} aflat pe o linie marcată și egal cu c_{rs} se adaugă matricei M o nouă linie formată din valorile i, s, c_{ts} și se marchează elementele c_{ts}, c_{st} din matricea C, după care se trece la E6.

E6) Se trece la etapa E4 sau se opresc calculele după cum mai există sau nu linii în matricea C cu nici un element nemarcat.

Dacă pentru matricea M rezultă un număr de linii mai mare decît ordinul grafului considerat, problema arborelui minim are mai multe soluții. În acest caz se permută liniile matricei M astfel încît elementele coloanei a doua să fie ordonate crescător. Pe coloana a doua se vor găsi numai indicii a n-1 vîrfuri ale grafului. Notînd frecvențele absolute ale acestor indici pe coloana a doua a matricei M cu F_i, iar frecvențele cumulate cu F_i^*; (i=1, 2, ..., n−1), numărul de arbori minimi ai grafului n_a are valoarea dată de relația:

$$n_a = \prod_{i=1}^{n-1} F_i \tag{5.33}$$

Pentru identificarea arborilor minimi se construiește o matrice A cu n-1 linii și n_a coloane, în etapele:

E7) Se asociază fiecărei frecvențe absolute F_i o variabilă $V_{i,k}$ ale cărei valori sînt elementele mulțimii $N_i=(F^*_{i-1}, F_i] \cap \{N\}$; (i=1, ..., n−1), iar F_0^*=0.

E8) Pe fiecare linie i a matricei A se înregistrează de n_a/F_i ori elementele $V_{i,k}$ ale mulțimii N_i, astfel ca matricea A obținută în final să aibă coloanele ordonate lexicografic.

Pe fiecare coloană a matricei A se găsesc indicii liniilor matricei ordonate M ce conține caracteristicile muchiilor unuia din arborii optimi.

În figura 5.3 se prezintă schema logică a programului ordinator OTREDIRA, elaborat pe baza algoritmului descris, unde: N este ordinul grafului atașat rețelei; $C(I, J)$ — matricea criterială asociată grafului topologic al rețelei; MIN(I) — minimul elementelor de pe linia I; NEM(I) — numărul elementelor minime de pe linia I; MAR(I) — numărul elementelor marcate de pe linia I; MINLM — minimul elementelor marcate de pe liniile ce conțin elemente marcate; $A(I, J)$ — matrice ajutătoare la identificarea arborilor minimi; $M(K, L)$ — matricea vîrfurilor muchiilor optime și a valorilor corespunzătoare.

În cazul că soluția obținută este multiplă, alegerea soluției optime se face ținînd seama și de alte criterii.

Folosirea algoritmului de optimizare expus conduce la economie de conducte, terasamente și energie electrică de pompare, precum și la o distribuție mai omogenă a debitelor și presiunii în rețea.

Scanned with OKEN Scanner

5.2.2. MODELE DE OPTIMIZARE PENTRU ECHILIBRAREA HIDRAULICĂ A REȚELELOR INELARE

5.2.2.1. Bazele calculului hidraulic

Atît la dimensionarea cît și la verificarea funcționării rețelelor de distribuție, calculul hidraulic este de cea mai mare însemnătate, avînd drept scop determinarea capacității de transport a conductelor, precum și calculul pierderilor de sarcină și al cotelor liniilor de presiune în lungul conductelor și a instalațiilor din sistem. La rețelele de tip inelar, acest calcul implică în ambele cazuri echilibrarea debitelor în rețea.

Oricărei rețele de distribuție îi corespunde un graf orientat asociat, compus din arce care pot reprezenta tronsoane, pompe sau armături și vîrfuri ca rezervoare, pompe sau intersecții de conducte, iar topologia unei astfel de rețele poate fi descrisă complet și univoc cu ajutorul matricei de incidență și a matricei de apartenență construite pentru graful respectiv.

Asociind unei rețele cu T tronsoane, N noduri și M inele graful topologic corespunzător, pe baza ecuației lui Euler (4.64) care dă numărul ciclomatic al unui graf, se poate scrie relația (4.86) pentru determinarea numărului maxim de inele independente.

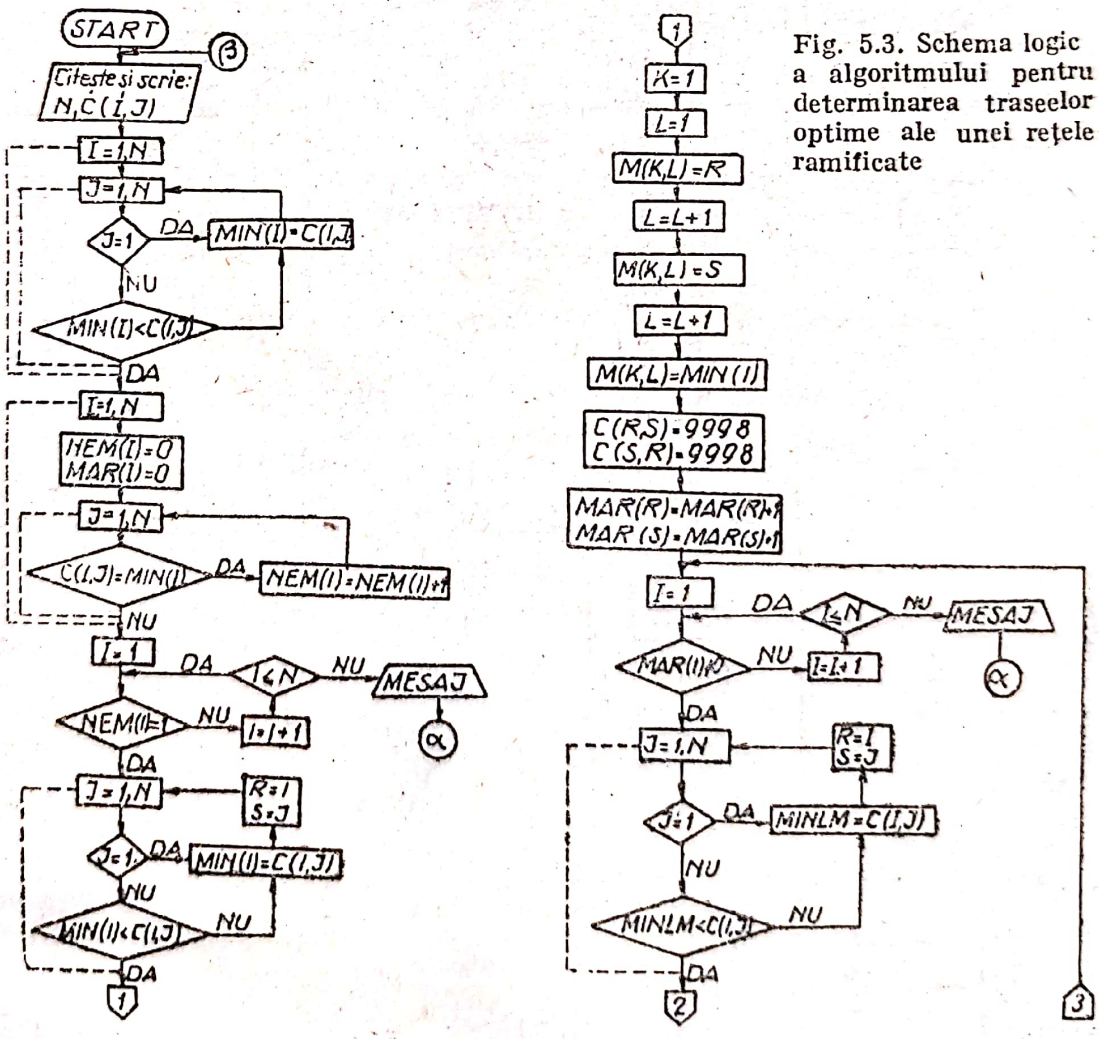


Fig. 5.3. Schema logic a algoritmului pentru determinarea traseelor optime ale unei rețele ramificate

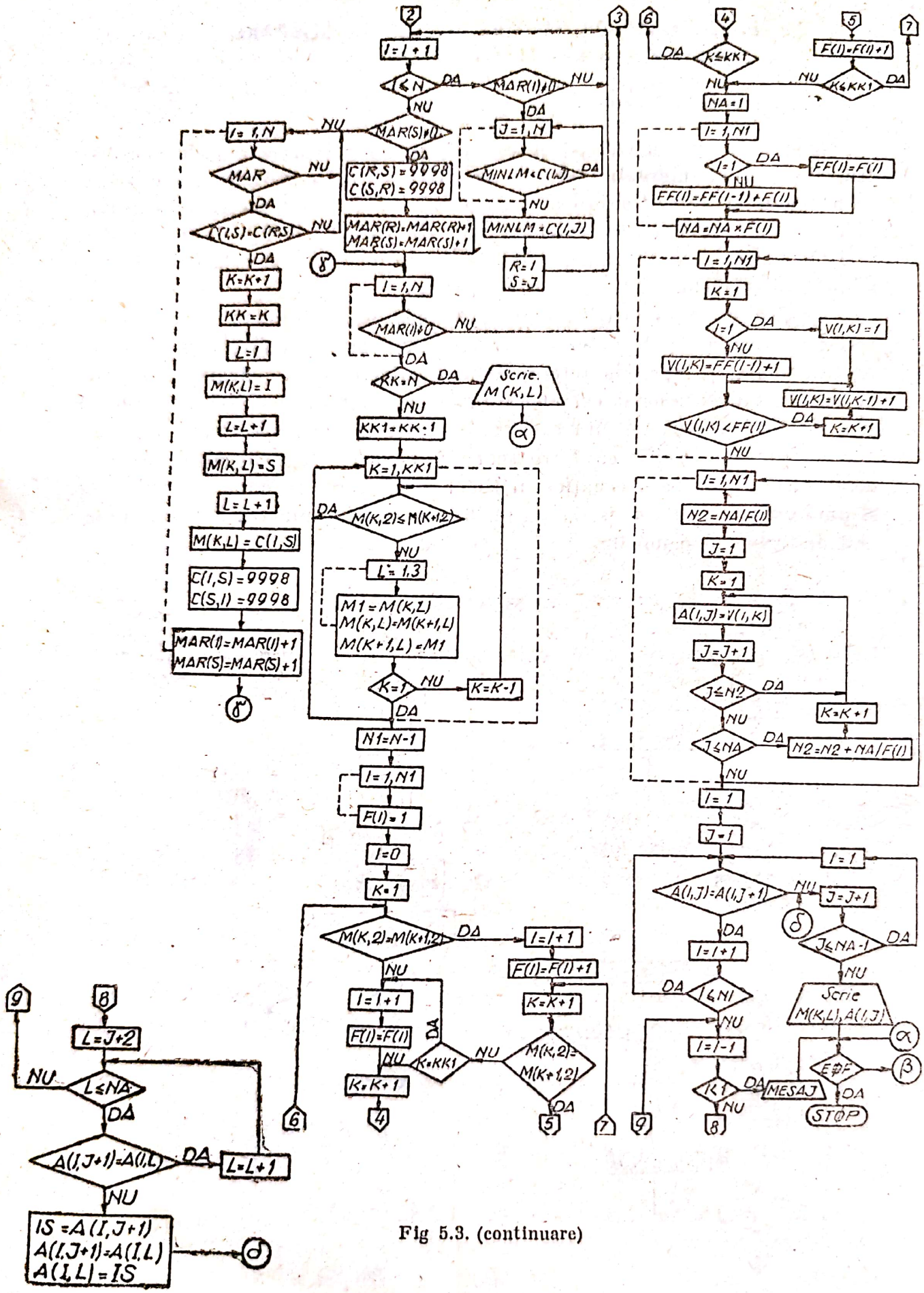


Fig 5.3. (continuare)

Scanned with OKEN Scanner

Pentru echilibrarea hidraulică a unei rețele inelare trebuie să se respecte legile de mișcare a apei de-a lungul tuturor conductelor, exprimate prin condițiile:

— continuitatea debitului în noduri:

$$\sum_{\substack{i=1\\ i\neq j}}^{N} Q_{ij} + q_j = 0 \qquad (j=1, \ldots, N-1) \tag{5.34}$$

în care: Q_{ij} este debitul tronsonului ij cu semnul (+) cînd intră în nodul j respectiv cu semnul (—) cînd pleacă din nodul j; q_j — debitul concentrat al nodului j cu semnul (+) cînd alimentează nodul și cu semnul (—) cînd se consumă în nod;

— conservarea energiei pe inele:

$$\sum_{\substack{ij=1\\ ij\in m}}^{T} \varepsilon_{ij} h_{ij} = 0 \qquad (m=1, \ldots, M) \tag{5.35}$$

în care: ε_{ij} este orientarea tronsonului ij cu valoarea (+1) sau (—1) după, cum sensul de curgere a apei în conducta ij este același sau opus cu sensul de parcurs al inelului; h_{ij} — pierderea longitudinală de sarcină pe tronsonul ij.

În cazul particular al rețelelor ramificate ($M=0$, $T=N-1$) numărul ecuațiilor de nod (5.34) egal cu $N-1$ este suficient pentru determinarea debitelor ca necunoscute.

Relațiile (5.34) și (5.35) constituie modelul matematic pe baza căruia s-au elaborat mai multe metode de echilibrare hidraulică a rețelelor inelare, în cadrul cărora se folosesc diverse procedee de rezolvare a sistemului de ecuații neliniare formulat, în funcție de mărimile hidraulice adoptate ca necunoscute.

Indiferent de metoda utilizată, se dispune de un număr T de ecuații, dat de relația (4.86) care pot avea ca necunoscute debitele sau pierderile de sarcină pe tronsoane. Astfel s-au conturat trei metode principale:

— metoda ciclurilor, care consideră ca necunoscute debitele de corecție pe inele;

— metoda nodurilor, care consideră ca necunoscute cotele piezometrice în noduri;

— metoda tronsoanelor, considerînd drept necunoscute debitele de tranzit pe tronsoane.

În metoda ciclurilor și în cea a nodurilor se pot aplica pentru rezolvarea sistemului neliniar de ecuații atît procedee iterative (Lobacev, Cross, Newton-Raphson), cît și procedee moderne de optimizare, folosind formulări variaționale, iar în metoda tronsoanelor se poate aplica pe lîngă procedeul iterativ de liniarizare a ecuațiilor de inel și un procedeu de optimizare care minimizează conținutul energetic al rețelei (modelul „conținut").

5.2.2.2. Relația funcțională pierdere de sarcină — debit în conductele rețelei

Potrivit prevederilor STAS-ului 4163-88, calculul pierderii de sarcină (energie) în conductele rețelelor exterioare de alimentare cu apă, în ipoteza debitului constant pe tronsoane, se efectuează cu relația Darcy-Weisbach:

$$h_{ij} = J_{ij} L_{ij} = \frac{8}{\pi^2 g} \lambda_{ij} \frac{L_{ij}}{D_{ij}^5} Q_{ij}^2 \tag{5.36}$$

Scanned with OKEN Scanner

în care: L_{ij}, D_{ij}, Q_{ij}, λ_{ij} sînt respectiv lungimea, diametrul, debitul și coeficientul de rezistență hidraulică al tronsonului ij, ce este funcție de regimul de curgere a apei în conductă; r — exponent cu valoarea 5,0; g — accelerația gravitațională; J_{ij} — pierderea de sarcină specifică pe tronsonul ij.

Întrucît relația (5.36) este incomod de utilizat pentru calculul unei rețele de conducte, se pune sub o formă generală asemănătoare cu cea pentru cazul curgerii apei în regim pătratic de tip Chézy-Manning:

$$h_{ij}=\frac{8\lambda_*}{\pi^2 g}\frac{L_{ij}}{D_{ij}^r}Q_{ij}^\beta=S_{ij}Q_{ij}^\beta \tag{5.37}$$

unde:

$$S_{ij}=k\frac{L_{ij}}{D_{ij}^r} \tag{5.38}$$

$$k=\frac{8\lambda_*}{\pi^2 g} \tag{5.39}$$

în care: S_{ij} este rezistența hidraulică a tronsonului ij; λ_*, β — valori medii constante ale coeficientului de rezistență hidraulică al conductelor și exponentului debitului, corespunzător intervalului de variație a debitelor tranzitate în rețea, stabilite cu ajutorul valorilor calculate în [89] în funcție de diferite rapoarte Δ/D.

Se observă că practic relația de tip Chézy-Manning poate fi obținută din relația (5.37) pentru $r=5{,}33$, $\beta=2$, $\lambda_*=k(\pi^2 g/8)$, $k=10{,}33\,n^2$, unde n este coeficientul de rugozitate al conductelor.

5.2.2.3. Modelul „conținutul rețelei"

Dacă pentru rețelele inelare se folosesc în loc de relațiile (5.34), (5.35) și (5.37), relațiile (5.34), (5.37) și o funcție obiectiv ce exprimă conținutul energetic minim al rețelei, echilibrarea hidraulică a rețelei se poate face printr-un procedeu de optimizare denumit „*modelul conținut*".

Potrivit acestui model, echilibrarea hidraulică a unei rețele inelare trebuie să satisfacă condițiile:

— continuitatea debitului în noduri (5.34);

— relația funcțională pierdere de sarcină-debit (5.37), scrisă sub forma:

$$Z_i-Z_j=S_{ij}Q_{ij}^\beta \tag{5.40}$$

în care: S_{ij} este modulul de rezistență hidraulică al tronsonului ij; Z_i, Z_j, — cotele piezometrice în nodurile i și j.

— condiția de cotă piezometrică cunoscută:

$$Z_j=Z_j^* \tag{5.41}$$

în care cu j s-au notat nodurile cu cota piezometrică Z_j^* cunoscută.

Determinarea debitelor Q_{ij} pentru care are loc echilibrarea rețelei se realizează pe criteriul minimizării conținutului de energie al rețelei, exprimat analitic prin funcția obiectiv:

$$F_e=\sum_{ij=1}^{r}\left[\int_0^{Q_{ij}}(Z_i-Z_j)dQ_{ij}\right]-\sum_{j=1}^{N_*}\left(\int_0^{q_j}Z_j^*dq_j\right)\to\min \tag{5.42}$$

supusă la restricțiile de continuitate (5.34) și cele de nenegativitate pentru debite:

$$Q_{ij}\geqslant 0 \tag{5.43}$$

unde N_* este numărul nodurilor de alimentare cu cotele piezometrice Z_j^*.

Scanned with OKEN Scanner

Funcționalele din primul termen al relației (5.42) semnifică pierderea de putere (energie) pe tronsoanele rețelei atunci cînd aceasta transportă debitele rezultate din condițiile de continuitate, iar funcționalele din al doilea termen semnifică aportul de putere provenit din exteriorul sistemului.

Condiția (5.43) arată că sensul de curgere a apei trebuie să fie obligatoriu cel ales în scrierea ecuațiilor de continuitate.

Problema exprimată prin relațiile (5.42), (5.43), (5.34) constă în găsirea unui set de debite care satisfac conservarea debitului și minimizează conținutul energetic al sistemului.

Ținînd seama de relația funcțională (5.40), în urma efectuării integralelor din (5.42), expresia funcției obiectiv devine:

$$F_e=\frac{1}{\beta+1}\sum_{ij=1}^{T} S_{ij}Q_{ij}^{\beta+1}-\sum_{j=1}^{N_*} Z_j^* q_j \rightarrow \min \tag{5.44}$$

care împreună cu restricțiile (5.34) și (5.43) formează un model de programare neliniară, ce poate fi rezolvat aplicînd agloritmul gradienților condiționați.

5.2.2.4. Model de echilibrare folosind formulări variaționale

Simplificări ale ecuațiilor fundamentale de calcul. În formularea clasică a problemei echilibrării hidraulice a rețelelor inelare, ecuațiile fundamentale de calcul sînt (5.34), (5.35) și (5.37).

Relația (5.37) se poate pune și sub formele:

$$Q_{ij}=S_{ij}^{-1/\beta}h_{ij}^{1/\beta}=S_{ij}^{-1/\beta}(Z_i-Z_j)\,|\,Z_i-Z_j\,|^{(1-\beta)/\beta} \tag{5.45}$$

$$h_{ij}=S_{ij}Q_{ij}\,|\,Q_{ij}\,|^{\beta-1} \tag{5.46}$$

$$Z_i-Z_j=S_{ij}Q_{ij}\,|\,Q_{ij}\,|^{\beta-1} \tag{5.47}$$

în care Z_i și Z_j sînt cotele piezometrice în nodurile i și j..

Asupra sistemului (5.34), (5.35), (5.37) de $2T$ ecuații cu $2T$ necunoscute se pot opera anumite simplificări, cu scopul de a reduce numărul variabilelor.

Astfel, introducînd relația (5.45) în ecuația (5.34) se obține sistemul de N-1 ecuații nodale echivalente cu N-1 necunoscute:

$$\sum_{\substack{i=1\\ i\neq j}}^{N} S_{ij}^{-1/\beta}(Z_i-Z_j)\,|\,Z_i-Z_j\,|^{(1-\beta)/\beta}+q_j=0 \qquad (j=1,\ldots,N\text{-}1) \tag{5.48}$$

Înlocuind relația (5.46) în ecuația (5.35) se obține sistemul:

$$\begin{aligned}&\sum_{\substack{i=1\\ i\neq 1}}^{N} Q_{ij}+q_j=0 \qquad (j=1,2,\ldots,N-1)\\ &\sum_{\substack{ij=1\\ ij\in m}}^{T} \varepsilon_{ij}S_{ij}Q_{ij}\,|\,Q_{ij}\,|^{\beta-1}=0 \qquad (m=1,\ldots,M)\end{aligned} \tag{5.49}$$

în care: $\varepsilon_{ij}=+1$ sau -1 după cum sensul de curgere a apei pe tronsonul ij este același sau opus cu sensul de parcurs al inelului m.

Scanned with OKEN Scanner

Asociind fiecărui inel m un debit de corecție ΔQ_m și alegînd o distribuție inițială de debite pe tronsoane $Q_{ij}^{(0)}$, care trebuie însă să satisfacă ecuațiile (5.34), se poate scrie:

$$Q_{ij}=Q_{ij}^{(0)}+\sum_{\substack{ij\in m\\ m=1}}^{M}\varepsilon_{ij}\Delta Q_m \qquad (ij=1,\ \ldots,\ T) \tag{5.50}$$

iar sistemul (5.49) se reduce la sistemul:

$$\sum_{\substack{ij\in m\\ ij=1}}^{T}\varepsilon_{ij}S_{ij}\Big(Q_{ij}^{(0)}+\sum_{\substack{ij\in m\\ m=1}}^{M}\varepsilon_{ij}\Delta Q_m\Big)\Big|Q_{ij}^{(0)}+\sum_{\substack{ij\in m\\ m=1}}^{M}\varepsilon_{ij}\Delta Q_m\Big|^{\beta-1}=0 \tag{5.51}$$

$$(m=1,\ \ldots,\ M)$$

În formularea (5.48) sistemul are N-1 grade de libertate, fiecare nod de cotă piezometrică necunoscută introducînd un grad de nedeterminare. Sistemul (5.51) are M grade de libertate. Față de o rețea ramificată ($T=N$-1), fiecare arteră suplimentară introduce un grad de nedeterminare în distribuția de debite pe tronsoane.

Formularea variațională a problemei. Se pornește de la modelul de echilibrare hidraulică *„conținutul rețelei"*, potrivit căruia se determină debitele de tranzit pe tronsoane astfel încît să minimizeze conținutul energetic al sistemului. În cadrul modelului *„conținut"* se pot elimina restricțiile de nenegativitate (5.43) prin folosirea relației (5.47).

Astfel, introducînd relațiile (5.47), (5.50) în funcția obiectiv (5.42) și efectuînd integralele se elimină toate restricțiile, problema echilibrării hidraulice a rețelelor inelare, în *metoda ciclurilor*, reducîndu-se la găsirea minimului unei funcții de M variabile, fără restricții:

$$F_e(\Delta Q)=\frac{1}{\beta+1}\sum_{ij=1}^{T}S_{ij}\Big|Q_{ij}^{(0)}+\sum_{\substack{ij\in m\\ m=1}}^{M}\varepsilon_{ij}\Delta Q_m\Big|^{\beta+1}-\sum_{j=1}^{N_*}Z_j^*q_j\rightarrow\min \tag{5.52}$$

De la o funcție cu T necunoscute și N-1 restricții se ajunge, deci, la o funcție doar cu M necunoscute fără restricții.

Se observă că punînd pentru funcția (5.52) condițiie de extremum $\partial F_e\,(\Delta Q)/\partial\Delta Q_m=0$ ($m=1,\ \ldots,\ M$), se obține sistemul de ecuații (5.51). Formularea variațională devine avantajoasă dacă, folosind un algoritm specific pentru a minimiza direct (5.52), acesta este mai eficient decît cei folosiți în formularea clasică.

Se poate ajunge la o formulare variațională a calculului de echilibrare hidraulică a rețelelor inelare și în *metoda nodală*. Astfel, folosind ca restricție ecuațiile (5.35) de conservare a energiei pe inele și ca funcție obiectiv:

$$F_e=\sum_{ij=1}^{T}\Big(\int_0^{h_{ij}}Q_{ij}\,dh_{ij}\Big)-\sum_{j=1}^{N_*}\Big(\int_0^{Z_j^*}q_j\,dZ_j\Big)\rightarrow\min. \tag{5.53}$$

în urma introducerii în (5.53) a relației funcționale (5.45) și a efectuării integralelor se elimină restricțiile, obținîndu-se:

$$F_e(Z)=\frac{\beta}{\beta+1}\sum_{ij=1}^{T}S_{ij}^{-1/\beta}\,|\,Z_i-Z_j\,|^{(\beta+1)/\beta}-\sum_{ij=1}^{N_*}q_jZ_j^*\rightarrow\min. \tag{5.54}$$

Scanned with OKEN Scanner

Punînd condițiile de extremum $\partial F_e(Z)/\partial Z_j=0$ $(j=1, \ldots, N)$ se obține sistemul de ecuații (5.48).

Aceste transformări reduc considerabil dimensiunile problemei, micșorîndu-se și efortul de programare cu calculatoarele electronice.

Formulările variaționale ale echilibrării hidraulice a rețelelor inelare devin avantajoase folosind un algoritm specific pentru a minimiza direct funcțiile (5.52) sau (5.54) cum este algoritmul gradienților conjugați.

Modelul de echilibrare folosind formulări variaționale are elemente comune atît ca metoda ciclurilor sau metoda nodală în formularea clasică, în sensul că se folosesc ca necunoscute debitele pe inele, respectiv cotele piezometrice în noduri, cît și cu modelul „conținut", problema fiind formulată ca minimul unei funcții.

Față de modelul „conținut" prezintă simplificări cu efecte favorabile asupra timpului de calcul și zonei de memorie. Astfel, de la o problemă de programare neliniară cu T necunoscute și N-1 restricții, avînd în plus cerința de a cunoaște aprioric sensul de curgere pe fiecare tronson se ajunge la o problemă de minim pentru o funcție doar cu M necunoscute, fără restricții.

Modelul de echilibrare în formulare variațională, generalizat pentru rețele inelare alimentate din mai multe surse și cu pompe integrate pe artere, s-a implementat în cadrul programelor ordinatoare ACIREV și ANOREV pentru analiza ciclică și respectiv nodală a acestor rețele de distribuție a apei.

5.2.3. DIMENSIONAREA OPTIMALĂ A REȚELELOR INELARE COMPLEXE

5.2.3.1. Bazele calculului optimal de dimensionare hidraulică

La proiectarea rețelelor inelare s-au conturat două probleme principale de calcul hidraulic:

— *analiza rețelei*, avînd drept scop determinarea debitelor reale în fiecare tronson și verificarea cotelor liniilor piezometrice, considerînd ca date topologia rețelei, debitele concentrate în noduri și respectiv diametrele, pentru care s-au studiat unele modele de optimizare la pct. 5.2.2.;

— *sinteza rețelei*, presupunînd atît dimensionarea rețelei, care este necesar a se efectua după criterii tehnico-economice și energetice bine fundamentate (sinteza optimală), cît și verificarea hidraulică a acestora.

Pentru o rețea inelară avînd un singur nod cu cota piezometrică cunoscută, se poate stabili relația (4.86) între numărul de tronsoane T, noduri N și inele închise independente (conținînd eventual și pompe integrate pe artere) M, iar în cazul topologiei cu rezervoare și pompe în noduri, cînd de fapt printre cele N noduri există N_{RP} noduri a căror cote piezometrice sînt cunoscute, la numărul de inele închise dat de relația (4.86) se mai adaugă N_{RP-1} inele deschise (fictive), astfel încît numărul total de inele independente se determină cu relația:

$$M=T-N+N_{RP} \tag{5.55}$$

în care N_{RP} este numărul total al rezervoarelor și pompelor din noduri.

Fiecare inel deschis leagă un nod cu cotă piezometrică cunoscută (rezervor) sau cu o relație între debit și nivel piezometric determinată (stație de pompare) cu un alt nod cu cotă piezometrică cunoscută sau cu o relație între debit și nivel piezometric determinată.

Scanned with OKEN Scanner

În calculul de dimensionare al unei rețele inelare cu T tronsoane intervin două serii de necunoscute: T valori ale debitelor în conducte și T valori ale diametrelor conductelor, în total $2T$ necunoscute.

Pentru dimensionare există T relații hidraulice independente ce exprimă legile de mișcare a apei de-a lungul tuturor conductelor (5.34) și (5.35), apărînd din punct de vedere matematic o problemă nedeterminată, cu gradul de nedeterminare egal cu numărul ciclomatic al rețelei.

Nedeterminarea, în metoda clasică, se ridică determinîndu-se diametrele printr-un calcul de predimensionare, iar în metoda optimizării ar trebui să se ridice obținînd celelalte T ecuații prin anularea derivatelor parțiale ale expresiei analitice a unui criteriu de optimizare în raport cu diametrele conductelor. Deci, teoretic, ar exista posibilitatea determinării unei soluții absolut optime, însă, practic, din punct de vedere matematic, se întîmpină serioase dificultăți la obținerea unui minim pe cale analitică, așa cum se arată în continuare.

5.2.3.2. Criterii de optimizare a dimensionării rețelelor

Calculele de optimizare a diametrelor rețelelor de distribuție a apei presupun stabilirea unei funcții obiectiv, mono- sau multicriteriale, care prezintă prioritate din punct de vedere economico-energetic în etapa respectivă.

În cazul dimensionării optimale a rețelelor de conducte, se pot folosi criterii de cost sau energetice simple ori compuse, care țin seama de investiția necesară realizării rețelei, costul energiei de pompare, cheltuielile de exploatare, energia consumată etc.

Investiția necesară pentru o rețea, constituită din T conducte, se obține prin însumarea investițiilor necesare fiecărei conducte componente, cu relația:

$$C_c = \sum_{ij=1}^{T} (a + bD_{ij}^{\alpha}) L_{ij} \tag{5.56}$$

în care: a, b, α sînt coeficienți de cost depinzînd de materialul conductei; D_{ij}, L_{ij} — diametrul și lungimea tronsonului ij.

Investiția în stația de pompare are forma:

$$C_p = \frac{9{,}81}{\eta} f \sigma Q_p (\Sigma h_{ij} + H_0) \tag{5.57}$$

în care: η este randamentul global al instalației de pompare; f — costul de instalare a unității de putere; σ — factor supraunitar ce ține seama de rezerva de putere instalată; Q_p — debitul pompat; H_0 — componenta geodezică și de utilizare a înălțimii de pompare; h_{ij} — suma pierderilor de sarcină pe un traseu defavorabil, plecînd de la stația de pompare spre punctul obligat.

Costul anual al energiei de pompare se definește prin relația:

$$C_e = W_e e = \frac{9{,}81}{\eta} 730\, e \tau \sum_{1}^{12} \Phi_k Q_p (\Sigma h_{ij} + H_0) \tag{5.58}$$

în care: W_e este energia consumată pentru pomparea apei; e — tariful energiei electrice; $\tau = T_p/8760$ — coeficientul pompării ce ține seama de numărul efectiv de ore T_p de pompare pe perioada unui an; Φ_k — raportul dintre debi-

Scanned with OKEN Scanner

tul lunar mediu și debitul pompat, cu valoarea 1 pentru unități industriale la care se livrează un debit constant pe tot parcursul anului, iar pentru centre populate se poate opera cu un șir de 12 valori, căruia îi corespunde $\Sigma\Phi_k = 10{,}44$.

Cheltuielile anuale de exploatare se exprimă sub forma:

$$C_{ex} = p_1 C_c + p_2 C_p + C_e \tag{5.59}$$

în care: p_1, p_2 sînt cotele de repartiții, întreținere și revizii periodice pentru conducte, respectiv pentru stația de pompare.

Cheltuielile anuale de investiție și exploatare se definesc prin funcția multicriterială:

$$C_{an} = \beta_0 (C_c + C_p) + C_{ex} \tag{5.60}$$

în care $\beta_0 = 1/T_r$ este cota de amortizare pentru durata de exploatare T_r.

Cheltuielile totale actualizate se exprimă prin funcția:

$$C_{ac} = C_c + C_p + \frac{(1+\beta_0)^t - 1}{\beta_0 (1+\beta_0)^t} C_{ex} \tag{5.61}$$

și se consideră pe toată durata de exploatare ($t = T_r$).

Energia înglobată în rețea se exprimă sub forma binomială (5.56) în care parametrii a, b, α au valori determinate în mod corespunzător.

Consumul energetic al rețelei, însumînd energia înglobată în rețea și cea cheltuită pentru exploatarea rețelei:

$$W_t = (\beta_0 + p_1) W_c + W_e \tag{5.62}$$

în care: W_c este energia înglobată în rețea; W_e — energia consumată pentru pomparea apei, dedusă din relația (5.58).

Ținînd seama de relațiile (5.56) ... (5.62) și făcînd notațiile:

$$r_a = \frac{(1+\beta_0)^t - 1}{\beta_0 (1+\beta_0)^t} \tag{5.63}$$

$$\xi_1 = r_a p_1 + \frac{t}{T_r}; \quad \xi_2 = r_a p_2 + \frac{t}{T_r} \tag{5.64}$$

$$\psi = \frac{9{,}81}{\eta} \left(f \sigma \xi_2 + 730 r_a e \tau \sum_1^{12} \Phi_k \right) \tag{5.65}$$

se stabilește o funcție obiectiv complexă multicriterială de forma generală:

$$F_c = \xi_1 \sum_{ij=1}^{T} (a + b D_{ji}^{\alpha}) L_{ij} + \psi \sum_{j=1}^{N_p} Q_{pj} (\Sigma h_{ij} + H_0)_j \tag{5.66}$$

unde: t este durata pentru care se calculează criteriul de optimizare exprima prin funcția obiectiv, avînd valoarea 1 sau T_r; N_p — numărul stațiilor de pompare.

Funcția obiectiv (5.66) dă posibilitatea ca prin particularizarea parametrului de timp t și a celorlalți parametri economici și energetici să se obțină cîte o funcție particulară.

Pentru rețele alimentate prin pompare se recomandă folosirea criteriului cheltuielilor anuale minime, dar adoptînd la realizarea investiției diametrele optime astfel obținute, ele devin neeconomice la un moment dat după execuție datorită evoluției prețurilor în timp.

Scanned with OKEN Scanner

De aceea se impune fie dinamizarea criteriului menționat folosind criteriul cheltuielilor actualizate minime, primul reprezentînd doar un caz particular al celui de-al doilea, atunci cînd investiția se realizează într-un an, cheltuielile de exploatare sînt aceleași de la an la an, iar durata de viață a rețelei este mare, cît mai ales utilizarea unor criterii energetice diferite de cele de cost.

În acest sens, un alt mod de a pune problema, care-și păstrează mai bine valabilitatea în timp și conduce la omogenizarea funcției obiectiv este dimensionarea rețelei pe criteriul consumului energetic minim.

5.2.3.3. Modele de calcul al repartiției optime a debitelor de tranzit

Din punct de vedere matematic, funcția obiectiv (5.66) folosită la optimizarea dimensionării rețelelor inelare funcționînd prin pompare este o funcție atît de debitul Q_{ij} (prin intermediul pierderii de sarcină h_{ij}), cît și de diametrul D_{ij} ale fiecărui tronson ij, întîmpinîndu-se serioase dificultăți în problema obținerii unui minim pe cale analitică.

Prin anulare, derivatele parțiale de forma $\partial F_c/\partial D_{ij}$ și $\partial F_c/\partial Q_{ij}$ arată că valoarea criteriului F_c admite extreme, însă derivatele de ordinul doi arată că valoarea acestui criteriu trece prin minime în raport cu diametrele D_{ij}, care se pot astfel optimiza, însă prin valori maxime în raport cu debitele Q_{ij}, ceea ce complică soluționarea problemei.

De asemenea, la dimensionarea optimală a rețelelor inelare efectuată la necesarul mediu de apă, față de care se cunosc debitele concentrate în nodurile rețelei, debitele de tranzit se pot stabili într-o infinitate de moduri, astfel încît să se satisfacă condițiile (5.34) de continuitate în noduri, afectînd însă gradul de siguranță și condițiile tehnico-economice ale sistemului.

De aceea, calculul de dimensionare optimală a rețelelor inelare este necesar a se efectua în două etape:

— optimizarea repartiției debitelor de tranzit pe baza unui criteriu calitativ care să țină seama de gradul de siguranță în exploatare;

— aplicarea unui criteriu de determinare efectivă a diametrelor optime pe baza debitelor optimizate.

Drept criteriu pentru optimizarea debitelor de tranzit se poate utiliza criteriul lucrului de transport minim, care se exprimă analitic prin M ecuații de forma:

$$F_t=\sum_{(m)} L_{ij} Q_{ij}^{\gamma} \rightarrow \min \quad (m=1, \ldots, M) \tag{5.67}$$

pentru fiecare inel m, unde lungimile L_{ij} ale conductelor sînt cunoscute.

Alături de cele M ecuații (5.67), modelul de optimizare a debitelor de tranzit conține și $N\text{-}N_{RP}$ ecuații de continuitate independente:

$$\sum_{\substack{i=1 \\ i\neq j}}^{N} Q_{ij}+q_j=0 \qquad (j=1, \ldots, N\text{-}N_{RP}) \tag{5.68}$$

Prin modelul (5.67), (5.68) nodurile rețelei sînt alimentate pe drumul cel mai scurt și cu un efort minim de transport, iar rezolvarea sa se efectuează aplicînd metoda iterațiilor, calculînd cîte un debit de corecție ΔQ_m pentru fiecare inel m, din condiția:

$$F_t=\sum_{(m)} L_{ij}(Q_{ij}+\Delta Q_m)^{\gamma} \rightarrow \min, \tag{5.69}$$

Scanned with OKEN Scanner

care presupune egalarea cu zero a derivatei $\partial F_t/\partial \Delta Q_m = f(\Delta Q_m)$, adică:

$$f(\Delta Q_m)=\gamma\sum_{(m)} L_{ij}(Q_{ij}+\Delta Q_m)^{\gamma-1}=0 \qquad (5.70)$$

de unde, pentru simplificare se reţin numai primii doi termeni ai seriei Mac Laurin:

$$f(\Delta Q_m)=f(0)+\Delta Q_m f'(0)=0 \qquad (5.71)$$

Făcînd derivata funcţiei (5.70) şi particularizînd-o pentru $\Delta Q_m=0$, rezultă:

$$f'(0)=\gamma(\gamma-1)\sum_{(m)} L_{ij}Q_{ij}^{\gamma-2} \qquad (5.72)$$

cu care, din relaţia (5.71), se obţine:

$$\Delta Q_m=-\frac{\sum_{(m)} L_{ij}Q_{ij}^{\gamma-1}}{(\gamma-1)\sum_{(m)} L_{ij}|Q_{ij}|^{\gamma-2}} \qquad (m=1, \ldots, M) \qquad (5.73)$$

unde debitele Q_{ij} sînt mărimi orientate, avînd semnul (+) atunci cînd sensul lor pe tronsoane coincide cu sensul de parcurs al inelului, iar în caz contrar avînd semnul (—).

Pornind de la soluţia iniţială se corectează debitele iterativ, pînă se atinge precizia prescrisă prin eroarea maximă admisă pe un ciclu.

Debitul corectat Q_{ij} se determină cu relaţia (5.74) sau (5.75) după cum conducta considerată este singulară sau comună la inelele m şi k:

$$Q_{ij}=Q_{ij}^{(0)}+\Delta Q_m \qquad (5.74) \qquad Q_{ij}=Q_{ij}^{(0)}+\Delta Q_m-\Delta Q_k \qquad (5.75)$$

în care: $Q_{ij}^{(0)}$ este debitul în conducta ij la aproximaţia precedentă; ΔQ_m, ΔQ_k — corecţia de debit ce se aplică pe inelul m şi respectiv k.

Pe baza acestui model de optimizare iterativ s-a elaborat subprogramul OPREDEC, care s-a implementat în cadrul unor programe ordinatoare complexe de analiză sau dimensionare a reţelelor inelare de distribuţie a apei.

Modelul de optimizare a debitelor de tranzit poate fi reformulat ca o problemă de minim condiţionat, avînd funcţia obiectiv unică:

$$F_t=\sum_{ij=1}^{T} L_{ij}Q_{ij}^{\gamma}\rightarrow \min., \qquad (5.76)$$

supusă la restricţiile:

$$Q_{ij}\geqslant Q_0^*; \qquad \sum_{\substack{i=1\\ i\neq j}}^{N} Q_{ij}+q_j=0 \qquad (j=1, \ldots, N\text{-}N_{RP}) \qquad (5.77)$$

în care Q_0^* este un debit minim necesar în conductele reţelei astfel încît să se evite posibilitatea transformării reţelei inelare într-una parţial ramificată, iar pentru rezolvarea acestei probleme se poate folosi metoda gradienţilor condiţionaţi.

Modelul de optimizare a debitelor (5.76), (5.77) are dezavantajul că necesită cunoaşterea apriorică a sensului de curgere pe fiecare tronson.

Scanned with OKEN Scanner

Din analizele numerice, efectuate pe diverse rețele inelare cu dimensiuni și încărcări hidraulice diferite, a rezultat valoarea optimă a exponentului γ din funcțiile obiectiv (5.67) și (5.76) ca fiind egală cu 2, așa încît se recomandă minimizarea produselor de transport pătratice pentru determinarea repartiției optime a debitelor de tranzit.

5.2.3.4. Model neliniar de optimizare a dimensionării rețelelor inelare

Modelul neliniar de optimizare face posibilă dimensionarea rețelelor inelare complexe prin minimizarea unuia dintre criteriile referitoare la costuri, energie înglobată, consumul energetic etc., exprimate prin funcția obiectiv de forma generală (5.66).

Dacă se explicitează diametrul D_{ij} din relația funcțională (5.37):

$$D_{ij}=k^{1/r}Q_{ij}^{\beta/r}h_{ij}^{-1/r}L_{ij}^{1/r} \tag{5.78}$$

și se introduce în funcția obiectiv (5.66), ținînd seama de ecuațiile de continuitate a debitului și de conservare a energiei pe inele și pe cîte un traseu de conducte pornind de la instalațiile de punere sub presiune la un punct obligat din sistem, dimensionarea optimală a rețelelor inelare alimentate prin pompare, se poate efectua cu modelul:

$$F_c=\xi_1\sum_{ij=1}^{T}(a+bk^{\frac{\alpha}{r}}Q_{ij}^{\frac{\beta\alpha}{r}}h_{ij}^{-\frac{\alpha}{r}}L_{ij}^{\frac{\alpha}{r}})L_{ij}+\psi\sum_{j=1}^{N_p}Q_{pj}(\Sigma h_{ij}+H_0)_j\rightarrow\min. \tag{5.79}$$

$$\sum_{\substack{i=1\\ i\neq j}}^{N}Q_{ij}+q_j=0 \qquad (j=1,\ \ldots,\ N\text{-}N_{RP}) \tag{5.80}$$

$$\sum_{\substack{ij=1\\ ij\in m}}^{T}\varepsilon_{ij}h_{ij}-f_m=0 \qquad (m=1,\ \ldots,\ M) \tag{5.81}$$

$$Z_{IPP,\,k}-\sum_{ij=1}^{NT_k}\varepsilon_{ij}h_{ij}-Z_n=0 \qquad (k=1\ ,\ldots,\ N_{RP}) \tag{5.82}$$

unde: $Z_{IPP,\,k}$ este cota piezometrică la instalația de punere sub presiune IPP_k; Z_n — cota piezometrică în punctul obligat n; NT_k — numărul tronsoanelor de conducte aflate pe traseul IPP_k-n; Q_{ij} — debitul de tranzit al tronsonului ij cu semnul (+) cînd intră în nodul j, respectiv (—) cînd pleacă din nodul j; q_j — debitul concentrat al nodului j cu semnul (—) cînd se consumă în nod și semnul (+) cînd alimentează nodul; ε_{ij} — orientarea tronsonului ij cu valoarea (+1) sau (—1), după cum sensul de curgere a apei în conducta ij este același sau opus cu sensul de parcurs al inelului m sau al traseului k; h_{ij} — pierderea longitudinală de sarcină pe tronsonul ij; f_m — sarcina piezometrică introdusă de elemente potențiale în inelul m, dată de relațiile:

— inele închise simple:

$$f_m=0 \tag{5.83}$$

— inele închise conținînd pompe integrate pe artere:

$$f_m=\sum_{(m)}\varepsilon_{ij}H_{p,\,ij} \tag{5.84}$$

Scanned with OKEN Scanner

— inele deschise cu stații de pompare sau rezervoare în noduri:

$$f_m = Z_{IPP,\,I} - Z_{IPP,\,E} \tag{5.85}$$

în care: $H_{p,\,ij}$ este înălțimea de pompare a pompei integrate pe artera ij la debitul Q_{ij}; $Z_{IPP,\,I}$, $Z_{IPP,\,E}$ — cotele piezometrice la instalațiile de punere sub presiune de la intrare, respectiv ieșire din ciclul fictiv.

Modelul (5.79)...(5.82) reprezintă o problemă de programare neliniară cu restricții de egalitate, ce se transformă într-un sistem de ecuații neliniare aplicînd procedeul coeficienților nedeterminați de tip Lagrange.

Se formulează funcția lagrangean Γ de forma:

$$\Gamma = F_c + \sum_{n=1}^{N-N_{RP}} \Lambda_n \Big(\sum_{\substack{i\neq j\\ i=1}}^{N} Q_{ij} + q_j\Big) + \sum_{m=1}^{M} \Lambda_m \Big(\sum_{\substack{ij\in m\\ ij=1}}^{T} \varepsilon_{ij} h_{ij} - f_m\Big) + \sum_{k=1}^{N_{RP}} \Lambda_k \Big(Z_{IPP,\,k} - \sum_{ij=1}^{NT_k} \varepsilon_{ij} h_{ij} - Z_n\Big) \tag{5.86}$$

unde Λ_n, Λ_m, Λ_k sînt multiplicatori Lagrange.

Soluția optimă a modelului dat de relațiile (5.79)...(5.82) se obține prin anularea derivatelor parțiale de ordinul întîi ale funcției Γ în raport cu $y_i \in \{Q_{ij}, h_{ij}\}$ și multiplicatorii Λ_n, Λ_m, Λ_k, iar prin eliminarea acestor multiplicatori sistemul se reduce la $2T+N_{RP}$ ecuații cu $2T+N_{RP}$ necunoscute (Q_{ij}, h_{ij}, $Z_{IPP,\,k}$) alcătuit din:

a) $N - N_{RP}$ ecuații nodale de tipul (5.80);

b) M ecuații de inel (5.81);

c) N_{RP} ecuații funcționale (5.82);

d) N-N_{RP} ecuații energetico-economice de nod de tipul:

$$\sum_{\substack{i\neq j\\ i=1}}^{N} Q^*_{ij} = \begin{cases} \text{— noduri alimentate prin pompare:} \\ -\dfrac{\psi}{A_0} Q_{pj} \quad (j=1, \ldots, N_P); \\ \text{—celelalte noduri:} \\ 0 \quad (j=N_p+1, \ldots, N\text{-}N_{RP}) \end{cases} \tag{5.87}$$

în care:

$$Q^*_{ij} = Q_{ij}^{\frac{\beta\alpha}{r}} h_{ij}^{-\frac{\alpha+r}{r}} L_{ij}^{\frac{\alpha+1}{r}} \tag{5.88}$$

$$A_0 = \frac{\alpha}{r} \xi_1 b k^{\frac{\alpha}{r}} \tag{5.89}$$

e) M ecuații energetico-economice de inel, avînd forma:

$$\sum_{\substack{ij\in m\\ ij=1}}^{T} H^*_{ij} = 0 \qquad (m=1, \ldots M) \tag{5.90}$$

în care:

$$H^*_{ij} = Q_{ij}^{\frac{\beta\alpha-r}{r}} h_{ij}^{-\frac{\alpha}{r}} L_{ij}^{\frac{\alpha+r}{r}} \tag{5.91}$$

În principiu, sistemul de ecuații (5.80), (5.81), (5.82) și (5.90) permite determinarea variabilelor Q_{ij} și h_{ij}, însă trebuie cercetată existența extremului funcției obiectiv F_c:

$$\frac{\partial^2 F_c}{\partial h_{ij}^2} = \frac{\alpha+r}{r} A_0 Q_{ij}^{\frac{\beta\alpha}{r}} h_{ij}^{-\frac{\alpha+2r}{r}} L_{ij}^{\frac{\alpha+r}{r}} \tag{5.92}$$

$$\frac{\partial^2 F_c}{\partial Q_{ij}^2} = \beta A_0 \frac{\beta\alpha-2r}{r} Q_{ij}^{\frac{\beta\alpha-2r}{r}} h_{ij}^{-\frac{\alpha}{r}} L_{ij}^{\frac{\alpha+r}{r}} \tag{5.93}$$

Scanned with OKEN Scanner

Întrucît $Q_{ij} \geqslant 0$, $h_{ij} \geqslant 0$ şi ţinînd seama că pentru valorile uzuale ale lui α (1,20...2,25) raportul $(\alpha+r)/r>0$, rezultă că $\partial^2 F_c \partial / h_{ij}^2 > 0$.

Pentru valorile practice luate de β (1,8 –2,0) raportul $(\beta\alpha - r)/r<0$, astfel încît rezultă că $\partial^2 F_c / \partial Q_{ij}^2 < 0$.

În consecinţă, în toate cazurile funcţia obiectiv F_c este convex-concavă în domeniul de definire şi prin urmare nu are extremum. Pentru stabilirea unui extremum trebuie să se cunoască un set de variabile (Q_{ij} sau h_{ij}).

Considerînd ca necunoscute variabilele h_{ij}, calculul de optimizare se conduce în două etape principale:

— deteminarea repartiţiei debitelor de tranzit Q_{ij}, aplicînd procedeul secţiunilor fictive şi optimizarea acesteia, folosind unul din modelele descrise la pct. 5.2.3.3.;

— determinarea pierderilor de sarcină h_{ij} prin rezolvarea sistemului de ecuaţii (5.81), (5.82), (5.87), funcţie de care se calculează diametrele optime D_{ij} cu relaţia (5.78), ce se rotunjesc la valori comerciale, recalcularea pierderilor de presiune cu relaţia (5.36) sau (5.37) efectuarea echilibrării hidraulice a reţelei cu procedee clasice sau utilizînd unul din modelele de optimizare expuse la pct. 5.2.2.3 şi 5.2.2.4.

Folosind notaţiile:

$$c_{ij} = \overbrace{Q_{ij}^{r}}^{\beta\alpha} \overbrace{L_{ij}^{r}}^{\alpha+r} \qquad (5.94) \qquad x = -\frac{\alpha+r}{r} \qquad (5.95)$$

ecuaţiile energetico-economice nodale (5.87) devin:

$$\sum_{\substack{i=1 \\ i \neq j}}^{N} c_{ij} h_{ij}^{x} = \begin{cases} -\dfrac{\psi}{A_0} Q_{pj} & (j=1, \ldots, N_p) \\ 0 & (j=N_p+1, \ldots, N-N_{RP}) \end{cases} \qquad (5.96)$$

şi împreună cu ecuaţiile (5.81), (5.82) formează un sistem neliniar care se rezolvă numeric aplicînd algoritmul Newton-Raphson.

Pentru stabilirea unei aproximante iniţiale a lui h_{ij}, se calculează, cu valorile optimizate Q_{ij}, pierderile de sarcină folosind relaţia:

$$h_{ij} = \Theta^{1/x} S f_{ij} \qquad (5.97)$$

unde:

$$\Theta = \frac{\psi}{A_0} Q_p \qquad (5.98) \qquad Sf_{ij} = \overbrace{Q_{ij}^{\alpha+r}}^{\alpha\beta} L_{ij} \qquad (5.99)$$

în care $Q_p = \Sigma Q_{pj}$ este debitul pompat în reţea.

Dacă sînt date iniţial pierderile de sarcină h_{ij}, se determină debitele Q_{ij}, prin rezolvarea sistemului de ecuaţii (5.80), (5.82), (5.90), funcţie de care se calculează diametrele optime tot cu relaţia (5.78).

La o dimensionare optimală, linia piezometrică a unui traseu de NT_k conducte, situate în aceeaşi zonă de presiune, trebuie să reprezinte o linie poligonală care să se apropie cît mai bine de forma optimă exprimată prin ecuaţia:

$$Z_j = Z_{IPP,\,k} - \left[1 - \left(1 - \frac{d}{\sum_{ij=1}^{NT_k} L_{ij}} \right)^{\frac{\beta\alpha}{\alpha+r}+1} \right] \sum_{ij=1}^{NT_k} h_{ij} \qquad (5.100)$$

în care: Z_j este cota piezometrică în nodul n; d — distanţa nodului j faţă de instalaţia de punere sub presiune k.

Pe baza modelului de optimizare neliniară, s-a elaborat programul ordinator OPNELIN, avînd schema logică din figura 5.4.

Scanned with OKEN Scanner

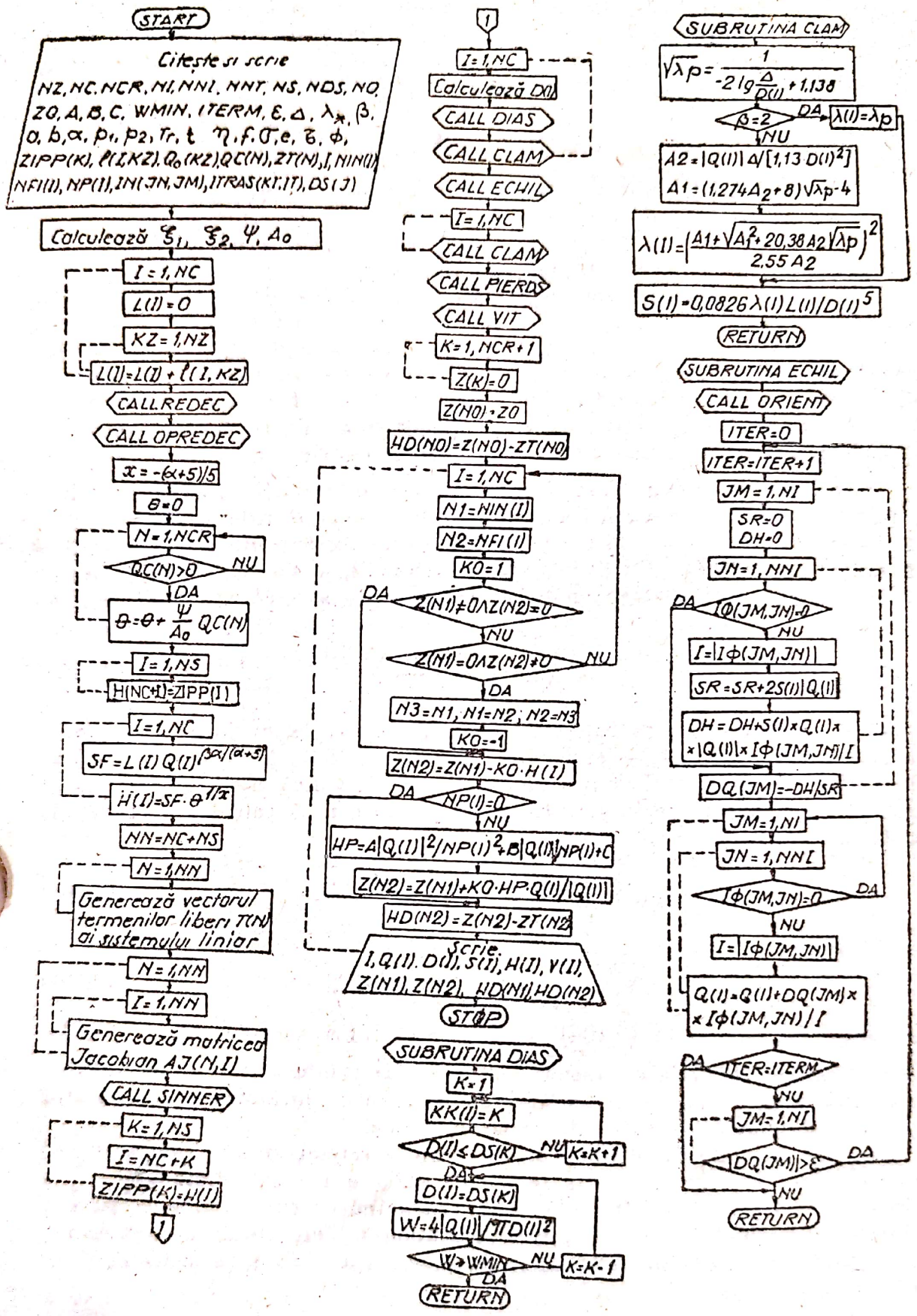


Fig. 5.4. Schema logică a modelului neliniar de optimizare a dimensionării rețelelor de distribuție a apei

Scanned with OKEN Scanner

Modelul de optimizare neliniară are un grad de generalitate și precizie sporit, fiind aplicabil rețelelor inelare alimentate din una sau mai multe surse, considerînd curgerea apei în regim turbulent tranzitoriu sau pătratic și oferind posibilitatea folosirii a multiple criterii de optimizare, putîndu-se alege în cadrul său ca clasă de variabile fie pierderile de sarcină, fie debitele în conducte.

5.2.3.5. Model liniar de optimizare a dimensionării rețelelor inelare

Pentru a evita dezavantajul modelului neliniar de optimizare de a altera valoarea optimă a diametrelor calculate ca urmare a alegerii unor diametre comerciale, se poate utiliza un model de dimensionare optimală a rețelelor noi sau parțial extinse funcționînd prin pompare sau gravitațional, ce se bazează pe metoda programării liniare și permite determinarea unei repartiții optime a diametrelor comerciale pe fiecare tronson al rețelei și a lungimii sectoarelor de tronsoane corespunzătoare acestor diametre, cu posibilitatea luării în considerare a consumului continuu uniform distribuit.

Generalizarea relației funcționale pierdere de sarcină specifică — debit la conductele rețelei cu consum continuu uniform distribuit. Admițînd că un tronson de conductă ij, parcurs de debitul Q_{ij}, este format din NS_{ij} sectoare k avînd diametrele $D_{k,ij}$ și ținînd seama de (5.36), se poate determina pentru fiecare sector k de pe tronsonul ij pierderea de sarcină specifică $J_{k,ij}$ cu relația:

$$J_{k,ij}=\frac{8}{\pi^2 g}\lambda_{k,ij}\frac{Q_{ij}^2}{D_{k,ij}^r} \tag{5.101}$$

unde: Q_{ij} este debitul de calcul pe tronsonul ij, corespunzător unei ipoteze de funcționare date; r — exponent cu valoarea 5,0; g — accelerația gravitațională; $\lambda_{k,ij}$ — coeficientul de rezistență hidraulică al sectorului k de pe tronsonul ij, care în regimul turbulenței tranzitorii se calculează cu relația explicită:

$$\sqrt{\lambda_{k,ij}}=\frac{A_2+\sqrt{A_2^2+20{,}38A_1\sqrt{\lambda_p}}}{2{,}55A_1} \tag{5.102}$$

cu notațiile:

$$\sqrt{\lambda_p}=\left(-2\lg\frac{\Delta}{D_{k,ij}}+1{,}138\right)^{-1} \tag{5.103}$$

$$A_1=Q_{ij}\Delta/(D_{k,ij}^2\nu) \tag{5.104}$$

$$A_2=(1{,}274\,A_1+8)\sqrt{\lambda_p}-4 \tag{5.105}$$

în care: Δ este rugozitatea absolută a peretelui conductei; ν — vîscozitatea cinematică a apei; λ_p — coeficientul de rezistență hidraulică corespunzător regimului turbulent pătratic de mișcare a apei.

Întrucît în realitate debitul pe tronsoanele rețelei (în speță de serviciu) scade de la o secțiune la alta în sensul de curgere a apei, acesta creează la ieșirea din conductă, printr-un fenomen asemănător reculului, o creștere a presiunii în conductă, care are ca efect diminuarea pierderii de sarcină. Astfel, ținînd seama de expresia modulului de rezistență hidraulică echivalent și introducînd notațiile:

$$\Theta_{ij}=\frac{4}{3}\frac{\theta_{ij}^2-3\theta_{ij}+3}{(2-\theta_{ij})^3} \tag{5.106}$$

$$\Omega_{k,ij}=\frac{4\alpha_0 D_{k,ij}}{\lambda_{k,ij}(2-\theta_{ij})} \tag{5.107}$$

unde: Q_{ij} este parametrul consumului pe tronsonul ij, avînd valoarea 1,00 dacă $Q_{ij} \leqslant Qa_{ij}/2$, iar în caz contrar avînd valoarea dată de relaţia (5.108) pentru tronsoanele reţelei ramificate de bază, care în cazul sistemelor ramificate coincide cu reţeaua considerată, sau de relaţia (5.109) pentru tronsoanele virtual suprimate din componenţa reţelei date, în cazul sistemelor inelare:

$$\theta_{ij} = \frac{Qa_{ij} + q_j}{Q_{ij} + \frac{Qa_{ij}}{2}} \qquad (5.108) \qquad \theta_{ij} = \frac{Qa_{ij}}{Q_{ij} + \frac{Qa_{ij}}{2}}, \qquad (5.109)$$

Qa_{ij} fiind debitul aferent tronsonului ij provenit din debitul distribuit; q_j — debitul concentrat în nodul final al tronsonului ij; α_0 — coeficientul de neuniformitate a distribuţiei vitezei în secţiunea conductei, expresia pierderii de sarcină specifică pe fiecare sector k al tronsonului ij de lungime $X_{k,ij}$, devine:

$$J^*_{k,ij} = J_{k,ij}\left(_{ij} - \Theta\frac{\Omega_{k,ij}}{X_{k,ij}}\right) \qquad (5.110)$$

Adoptînd în ipoteza consumului la noduri: $\theta_{ij} = \alpha_0 = 0$, rezultă $\Theta_{ij} = 1$; $\Omega_{k,ij} = 0$, astfel încît relaţia generalizată (5.110) primeşte forma particulară (5.101).

Formularea modelului matematic. Pentru sistemele inelare, se determină iniţial reţeaua optimă virtual ramificată de bază cu ajutorul algoritmului descris la pct. 5.2.1., iar pentru fiecare ipoteză de funcţionare se determină debitele de tranzit pe tronsoane Q_{ij}, a căror repartiţie se optimizează folosind modelul (5.73) cu $\gamma = 2$.

Se stabileşte pentru fiecare tronson ij seria diametrelor comerciale posibil a fi utilizate $D_{k,ij} \in [D_{max,ij}, D_{min,ij}]$ pe baza valorilor limită ale diametrelor optime $D_{max,ij}$ şi $D_{min,ij}$ calculate cu relaţia de optimizare (5.111), la reţelele alimentate prin pompare, sau cu relaţia (5.112), obţinută din condiţia ca viteza apei în conductă să fie cuprinsă între limitele vitezei economice V_{min} şi V_{max}, la reţelele alimentate gravitaţional:

$$D_{max(min),ij} = E^{\frac{1}{\alpha+r}}_{max(min)} Q_p^{\frac{1}{\alpha+r}} Q_{ij}^{\frac{\beta}{\alpha+r}} \qquad (5.111)$$

$$D_{max(min),ij} = \sqrt{\frac{4\,Q_{ij}}{\pi\, V_{min(max)\,ij}}} \qquad (5.112)$$

în care: debitul Q_{ij} este debitul de calcul al tronsonului ij; $Q_p = \Sigma Q_{pj}$ — debitul pompat în reţea.

Factorul economico-energetic al conductelor:

$$E = \frac{\psi\, r\, k}{\alpha\, b\, \xi_1} \qquad (5.113)$$

se introduce în (5.111) cu valoarea maximă şi respectiv minimă calculată în funcţie de criteriul de optimizare adoptat, corespunzător valorilor limită de variaţie ale parametrilor economico-energetici.

Pentru optimizarea diametrelor în cazul extinderii unei reţele existente, se foloseşte un coeficient de penalizare, p_{ij}, care la conductele cu diametre fixate are valoarea, în m, a diametrului corespunzător impus, rezultînd $D_{k,ij} = p_{ij}$.

Scanned with OKEN Scanner

Admiţînd că un tronson ij de lungime L_{ij} al unei reţele alcătuite din T tronsoane, funcţionînd prin pompare, poate fi format din NS_{ij} sectoare k de diametre $D_{k,ij}$ şi lungimi $X_{k,ij}$ şi ţinînd seama de notaţiile:

$$c^*_{k,ij}=\xi_1(a+bD^{\alpha}_{k,ij}) \qquad (5.114) \qquad Z_{IPP,j}=(\Sigma h_{ij}+H_0)_j \qquad (5.115)$$

funcţia obiectiv (5.66) primeşte forma:

$$F_c=\sum_{ij=1}^{T}\sum_{k=1}^{NS_{ij}}c^*_{k,ij}X_{k,ij}+\psi\sum_{j=1}^{N_p}Q_{pj}Z_{IPP,j}\rightarrow\min \qquad (5.116)$$

Necunoscutele funcţiei obiectiv sînt variabilele $X_{k,ij}$ şi $Z_{IPP,j}$ în număr de $N_p+\sum_{ij=1}^{T}NS_{ij}$.

În cazul că instalaţiile de punere sub presiune (IPP_j) sînt unul sau mai multe rezervoare ($\psi=0$), expresia (5.116) devine:

$$F_c=\sum_{ij=1}^{T}\sum_{k=1}^{NS_{ij}}c^*_{k,ij}X_{k,ij}\rightarrow\min \qquad (5.117)$$

minimizînd energia înglobată sau costul de investiţie al reţelei şi avînd ca necunoscute variabilele $X_{k,ij}$.

Aşadar, trebuie determinate valorile variabilelor, astfel încît să se minimizeze funcţia obiectiv dată în condiţiile satisfacerii următoarelor restricţii:

— de tip constructiv (C):

$$\sum_{k=1}^{NS_{ij}}X_{k,ij}=L_{ij} \qquad (ij=1,\ \ldots,\ T) \qquad (5.118)$$

— de tip funcţional (F), care se scriu în fiecare ipoteză de funcţionare, exprimînd condiţia ca pornind pe diverse trasee de la instalaţiile de punere sub presiune IPP_j (fig. 5.5) la punctele obligate n să fie asigurată presiunea necesară HN_n:

$$Z_{IPP,j}-\sum_{ij=1}^{NT_j}\sum_{k=1}^{NS_{ij}}\Theta_{ij}J_{k,ij}X_{k,ij}\geqslant ZT_n+HN_n-\sum_{ij=1}^{NT_j}\sum_{k=1}^{NS_{ij}}J_{k,ij}\Omega_{k,ij} \qquad (5.119)$$

în care: NT_j este numărul de tronsoane pe traseul IPP_j-n; ZT_n — cota terenului în nodul n; $Z_{IPP,j}$ — cota piezometrică la instalaţia de punere sub presiune j.

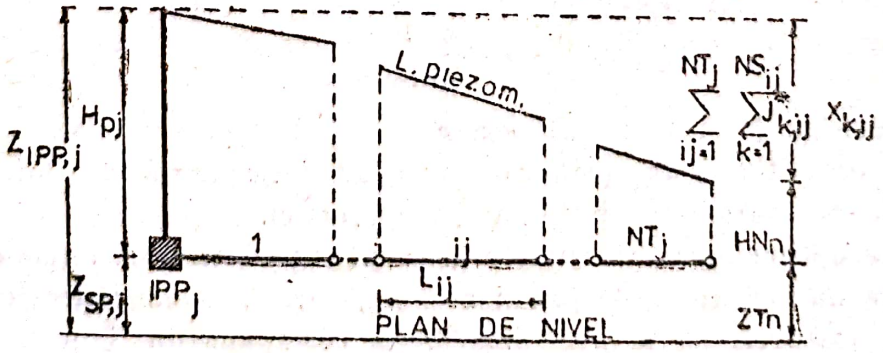


Fig. 5.5. Schema unui traseu IPP_j-punctul obligat n

Scanned with OKEN Scanner

— de tip hidraulic (H), caracteristice numai rețelelor inelare, exprimînd condiția de conservare a energiei pe fiecare inel m:

$$\sum_{(m)} \varepsilon_{ij} \Theta_{ij} J_{k,ij} X_{k,ij} = \sum_{(m)} \varepsilon_{ij} J_{k,ij} \Omega_{k,ij} + f_m \tag{5.120}$$

în care orientarea tronsoanelor ε_{ij} are valoarea $(+1)$ sau (-1), după cum sensul de curgere a apei în tronsonul ij este același sau opus cu sensul de parcurs al inelului m, iar sarcina piezometrică f_m este dată de una din relațiile (5.83), (5.84), (5.85).

În situația că nivelele piezometrice $Z_{IPP,j}$ sînt cunoscute, nefiind necesar a se determina prin optimizare, funcția obiectiv (5.116) primește forma (5.117), iar valorile $Z_{IPP,j}$ intră în componența termenului liber al restricțiilor (5.119) și respectiv (5.120).

Funcția obiectiv (5.116) sau (5.117) și restricțiile (5.118), (5.119), (5.120) fiind liniare în raport cu necunoscutele sistemului, soluția optimă se poate determina pe baza metodei programării liniare, utilizînd algoritmul Simplex.

Într-o soluție optimă, pentru un tronson dat, de obicei cel mult două din variabilele $X_{k,ij}$ vor fi diferite de zero, astfel încît soluția respectivă este realizabilă din punct de vedere tehnic, mai ales dacă lungimea tronsonului este însemnată.

După determinarea necunoscutelor $Z_{IPP,j}$ prin optimizare, la rețelele funcționînd prin pompare, rezultă înălțimile de pompare corespunzătoare:

$$H_{pj} = Z_{IPP,j} - Z_{SP,j} \tag{5.121}$$

în care $Z_{SP,j}$ este cota nivelului apei în bazinul de aspirație al IPP_j.

Ținînd seama de pierderea de sarcină H_n pe traseul IPP_j-n:

$$H_n = \sum_{ij=1}^{NT_j} \sum_{k=1}^{NS_{ij}} \Theta_{ij} J_{k,ij} X_{k,ij} - \sum_{ij=1}^{NT_j} \sum_{k=1}^{NS_{ij}} J_{k,ij} \Omega_{k,ij} \tag{5.122}$$

se obține cota piezometrică Z_n și presiunea disponibilă HD_n în nodul n, cu relațiile:

$$Z_n = Z_{IPP,j} - H_n \tag{5.123}$$

$$HD_n = Z_n - ZT_n \tag{5.124}$$

n care ZT_n este cota geodezică a nodului n.

În figura 5.6 se prezintă schema logică a modelului liniar de optimizare a diametrelor rețelelor de distribuție a apei, pe baza căruia s-a elaborat programul ordinator OPDIREDA.

Aplicarea modelului liniar de optimizare în ipoteza consumului uniform distribuit conduce la micșorarea consumului energetic și la o mai bună distribuție a energiei de pompare în sistem, prin eliminarea presiunilor disponibile sporite pe unele trasee chiar la consum maxim. Reducerea astfel a presiunii în rețea contribuie la micșorarea pierderilor de apă în sistem.

Scanned with OKEN Scanner

5.3. OPTIMIZAREA DIMENSIONĂRII REȚELELOR INTERIOARE DE ALIMENTARE CU APĂ ÎN CLĂDIRI

Practica actuală a dimensionării rețelelor interioare de alimentare cu apă rece și caldă se caracterizează printr-o alegere a diametrelor conductelor funcție de vitezele medii economice, care însă nu pot ține seama de dinamica parametrilor economico-energetici caracteristici acestora, fiind necesară optimizarea dimensionării lor.

În acest sens, se prezintă un model de calcul optimal pentru dimensionarea rețelelor cu distribuție interioară, care presupune cunoscute următoarele date de bază: felul și numărul consumatorilor de apă; schema coloanelor și planul de distribuție; lungimea, coeficientul de rezistență hidraulică locală și echivalentul de debit aferent pentru fiecare tronson de conductă; parame-

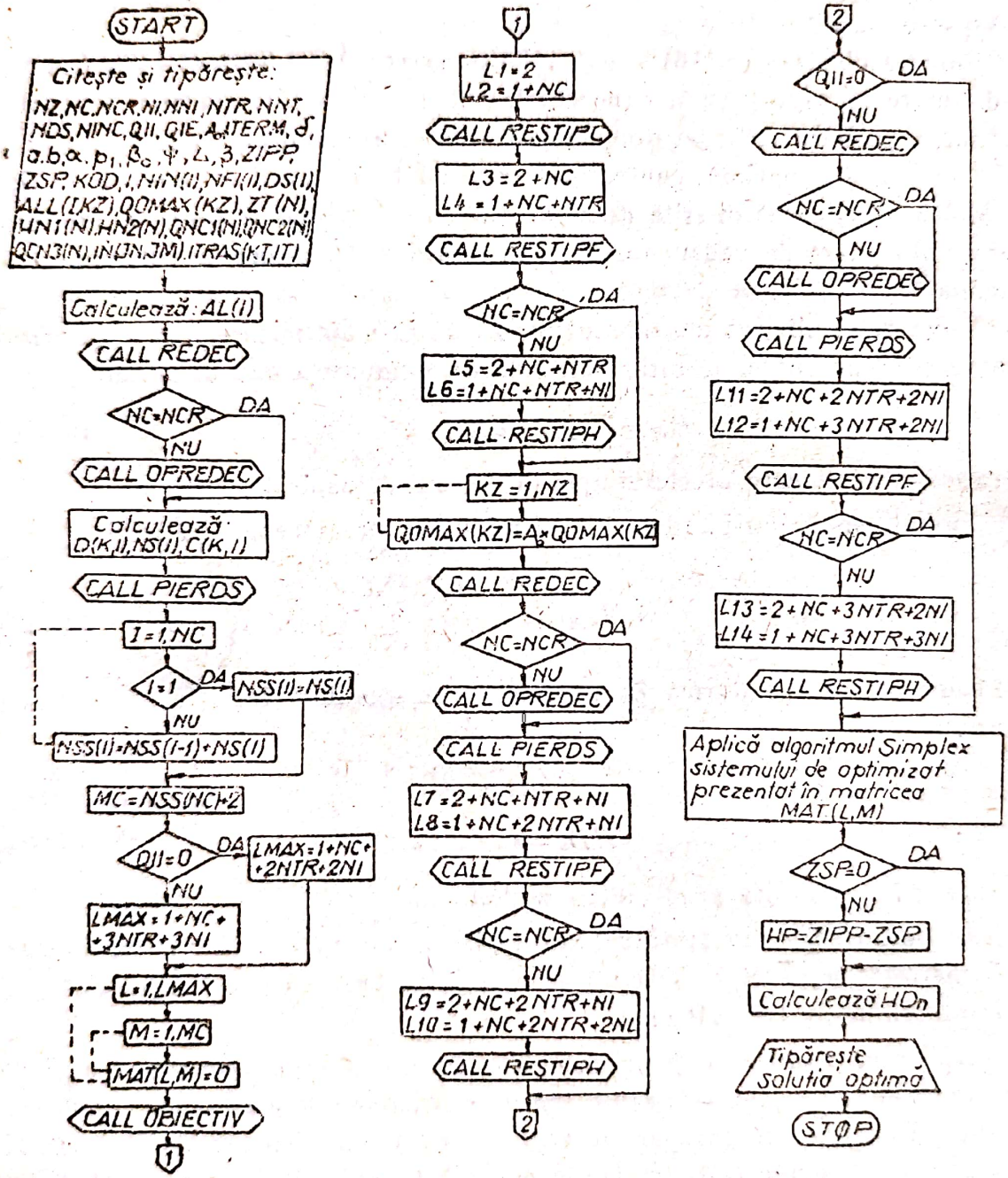


Fig. 5.6. Schema logică a modelului liniar de optimizare a dimensionării rețelelor de distribuție a apei

Scanned with OKEN Scanner

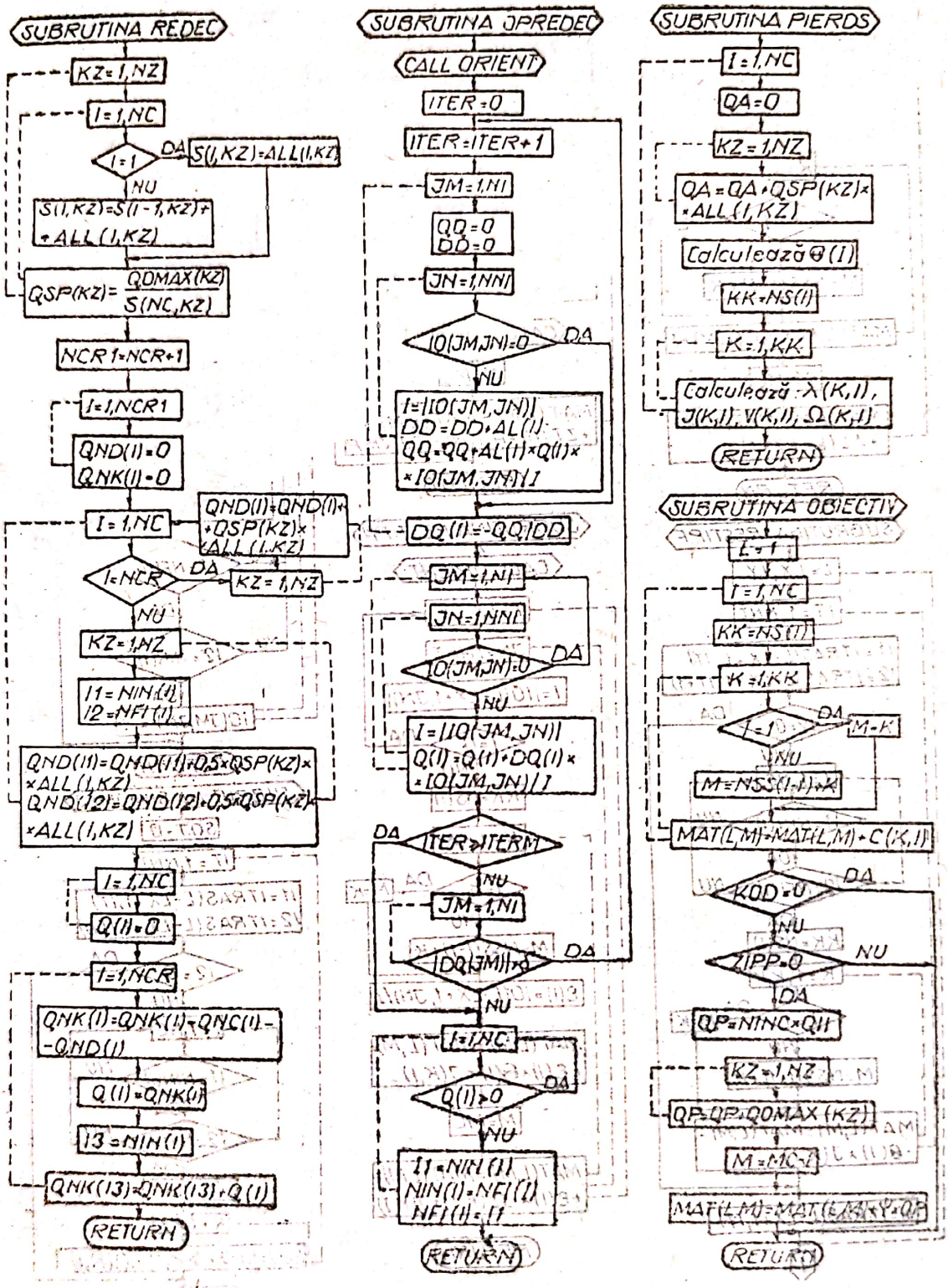


Fig. 5.6. (continuare)

Scanned with OKEN Scanner

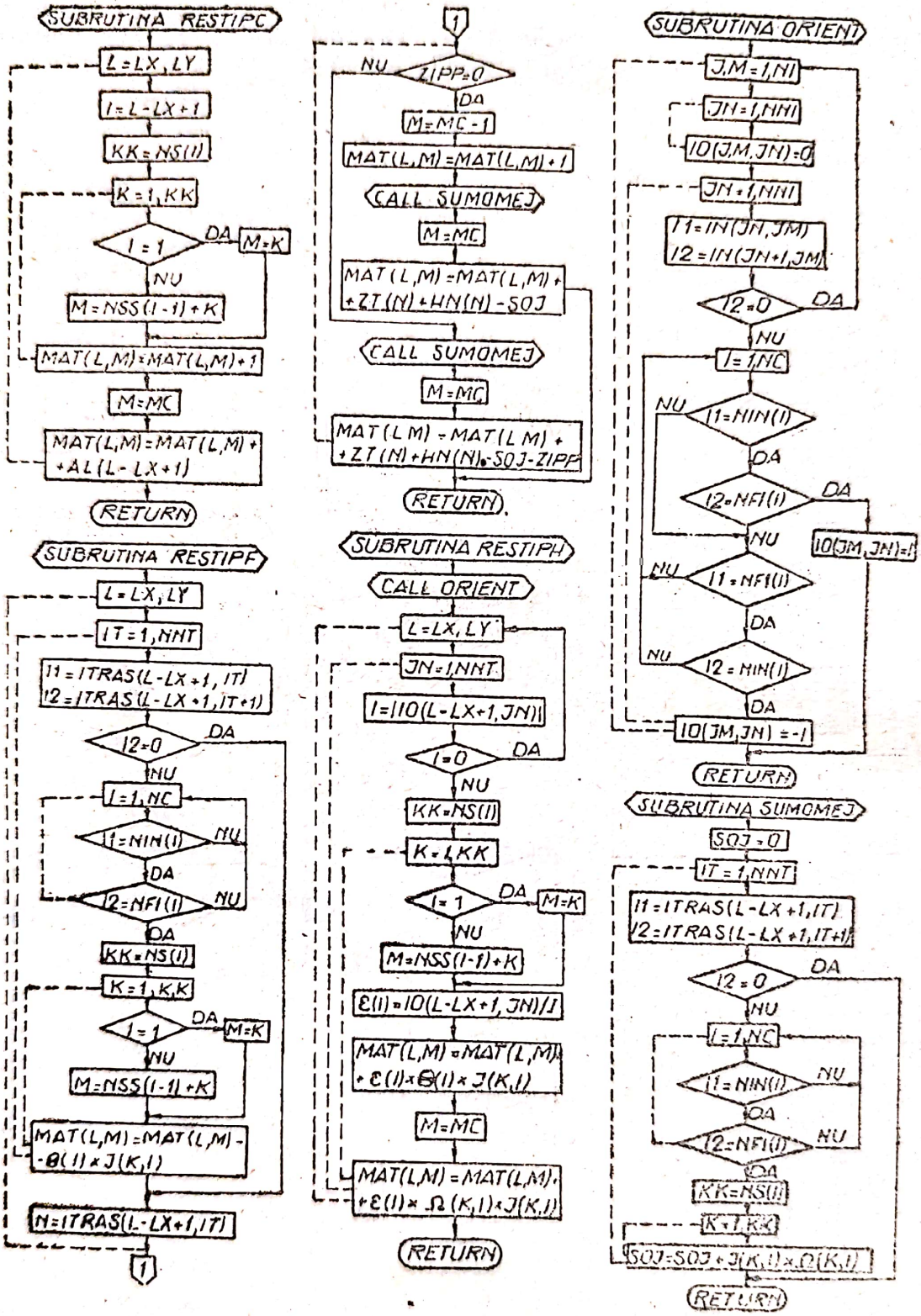


Fig. 5.6. (continuare)

trii economici și energetici; temperatura apei; înălțimile geodezice în fiecare nod al rețelei măsurate față de consumatorul situat în poziția cea mai defavorabilă pe fiecare coloană; presiunea de utilizare caracteristică fiecărei coloane.

După întocmirea schemei coloanelor și a planului de distribuție, se numerotează tronsoanele rețelei astfel încît nodul final al unui tronson se ia identic numărului de ordine al său, se începe cu 1 la nodul terminal al traseului ce conține punctul de consum din situația cea mai defavorabilă, iar apoi la fiecare ramificație se dă un număr nodului și tronsonului respectiv amonte, numai după ce au fost epuizate toate căile alimentate din ramificația considerată.

Se determină apoi repartiția debitelor de calcul pe tronsoane, folosind relația generală (5.125) pentru clădiri de locuit și relația (5.126) pentru clădiri social-culturale:

$$q_i = B(AC\sqrt{E_i} + 0{,}004\, E_i) \qquad (5.125) \qquad q_i = ABC\sqrt{E_i} \qquad (5.126)$$

în care: q_i este debitul de calcul al tronsonului i; E_i — suma echivalenților punctelor de consum alimentate de tronsonul i; A — coeficient funcție de regimul de furnizare a apei în rețeaua de distribuție; B — coeficient funcție de felul apei (rece sau caldă); C — coeficient funcție de destinația clădirii.

Pentru toate categoriile de clădiri la valori ale lui E_i mai mici decît cele indicate în STAS-ul 1478-90 se folosește relația:

$$q_i = ABE_i \qquad (5.127)$$

Avînd în vedere că pe întreaga durată de exploatare a rețelei se consumă energie necesară pentru pomparea apei, dimensionarea optimală a acesteia impune respectarea atît a condițiilor hidraulice de asigurare a debitului și presiunii de utilizare în toate punctele de consum, cît și a celor de natură economică, folosindu-se criteriul de optimizare a cheltuielilor anuale minime.

Ținînd seama de expresia pierderii de sarcină pe un tronson i:

$$h_i = \frac{8}{\pi^2 g}\frac{q_i^2}{D_i^4}\left(\lambda_i \frac{L_i}{D_i} + \zeta_i\right) = \frac{8}{\pi^2 g}\lambda_i \frac{q_i^2}{D_i^7} Le_i, \qquad (5.128)$$

de expresiile investiției în conductele rețelei, în stația de ridicare a presiunii și de costul anual al energiei electrice, funcția obiectiv are forma:

$$C_a = (p_1 + \beta_0)\sum_{i=1}^{T}(a + bD_i^{\alpha})L_i + (p_2 + \beta_0)\frac{9{,}81}{\eta} f\sigma Q(\Sigma h_i + H_0) + \\ + \frac{86\,000}{\eta} e\tau Q(\Sigma h_i + H_0) \qquad (5.129)$$

în care: D_i este diametrul tronsonului i cu lungimea L_i și lungimea echivalentă Le_i; λ_i — coeficientul de rezistență hidraulică longitudinală a tronsonului i; ζ_i — suma coeficienților de rezistență hidraulică locală a tronsonului i; r — exponent cu valoarea 5,0; Q — debitul total introdus în rețea; H_0 — componenta geodezică și de utilizare a înălțimii de pompare; $\beta_0 = 1/T_r$ — cota de amortizare pentru durata de exploatare T_r; a, b, α — parametrii costului specific al conductelor; p_1, p_2 — cotele de reparații, întreținere și revizii periodice pentru conducte, respectiv stația de ridicare a presiunii; η — randamentul general al instalației de pompare; f — costul de instalare a

Scanned with OKEN Scanner

unității de putere; σ — factor ce ține seama de rezerva de putere instalată; e — tariful energiei electrice; τ — coeficient ce ține seama de neuniformitatea funcționării stației de ridicare a presiunii.

Anulînd derivata parțială a funcției obiectiv (5.129) în raport cu oricare diametru D_i al rețelei și introducînd notațiile:

$$\psi=\frac{9,81}{\eta}[(p_2+\beta_0)f\sigma+8760e\tau] \qquad (5.130) \qquad E_0=\frac{8r\psi}{\pi^2 g\alpha b(p_{1+}\beta_0)} \qquad (5.131)$$

se obține expresia generală a diametrului optim:

$$D_i=(E_0Q)^{\frac{1}{\alpha+r}}\lambda_i^{\frac{1}{\alpha+r}}q_i^{\frac{2}{\alpha+r}} \qquad (5.132)$$

Ținînd seama că în regimuri normale de exploatare conductele rețelei lucrează în domeniul turbulenței tranzitorii, pentru calculul coeficientului λ_i se folosește relația (5.102), iar sistemul format din ecuațiile (5.102), (5.132) ce conține necunoscutele λ_i și D_i se rezolvă prin aproximații succesive.

Pentru traseul pe care se găsește punctul de consum din situația cea mai defavorabilă se determină presiunea necesară H_{nec} la punctul de alimentare, pentru a asigura buna funcționare a tuturor consumatorilor:

$$H_{nec}=\Sigma h_i+H_g+H_u \qquad (5.133)$$

în care: Σh_i este pierderea de sarcină pe traseul de conducte cel mai defavorabil; H_g — înălțimea geodezică; H_u — presiunea de utilizare la punctul de consum cel mai dezavantajat.

Pentru traseele pe care se găsesc celelalte puncte de consum se calculează presiunea disponibilă în limita căreia se stabilesc diametrele tronsoanelor respective, cu condiția nedepășirii vitezelor maxime admisibile în funcție de gradul de confort al clădirii. Disponibilul excedentar de presiune, calculat ca diferența dintre pierderea de sarcină de pe traseul cel mai defavorabil și pierderile de sarcină de pe traseul de alimentare a consumatorilor respectivi, se preia cu ajutorul diafragmelor de reglaj, al căror diametru, în mm, se determină cu relația:

$$D_0=\alpha_1\sqrt{\frac{q}{\Delta H^{0,5}}} \qquad (5.134)$$

în care: q este debitul vehiculat prin diafragmă, în dm³/s; ΔH — presiunea disponibilă excedentară, în mCA; iar:

$$\alpha_1=27,1-3,35\beta_1, \qquad (5.135)$$

unde β_1 reprezintă raportul între grosimea și diametrul diafragmei.

În figura 5.7 se prezintă schema logică a modelului de optimizare implementat în cadrul programului ordinator DIREINTO pentru dimensionarea rețelelor interioare de alimentare cu apă cu distribuție inferioară, în situația în care trebuie determinată sarcina hidrodinamică necesară, în punctul de alimentare, unde: ε este eroarea admisă pentru diferența de presiune în punctele de echilibrare (10...50 mmCA), iar ε_1 — eroarea admisă la calculul coeficienților λ_i (0,0001—0,00005).

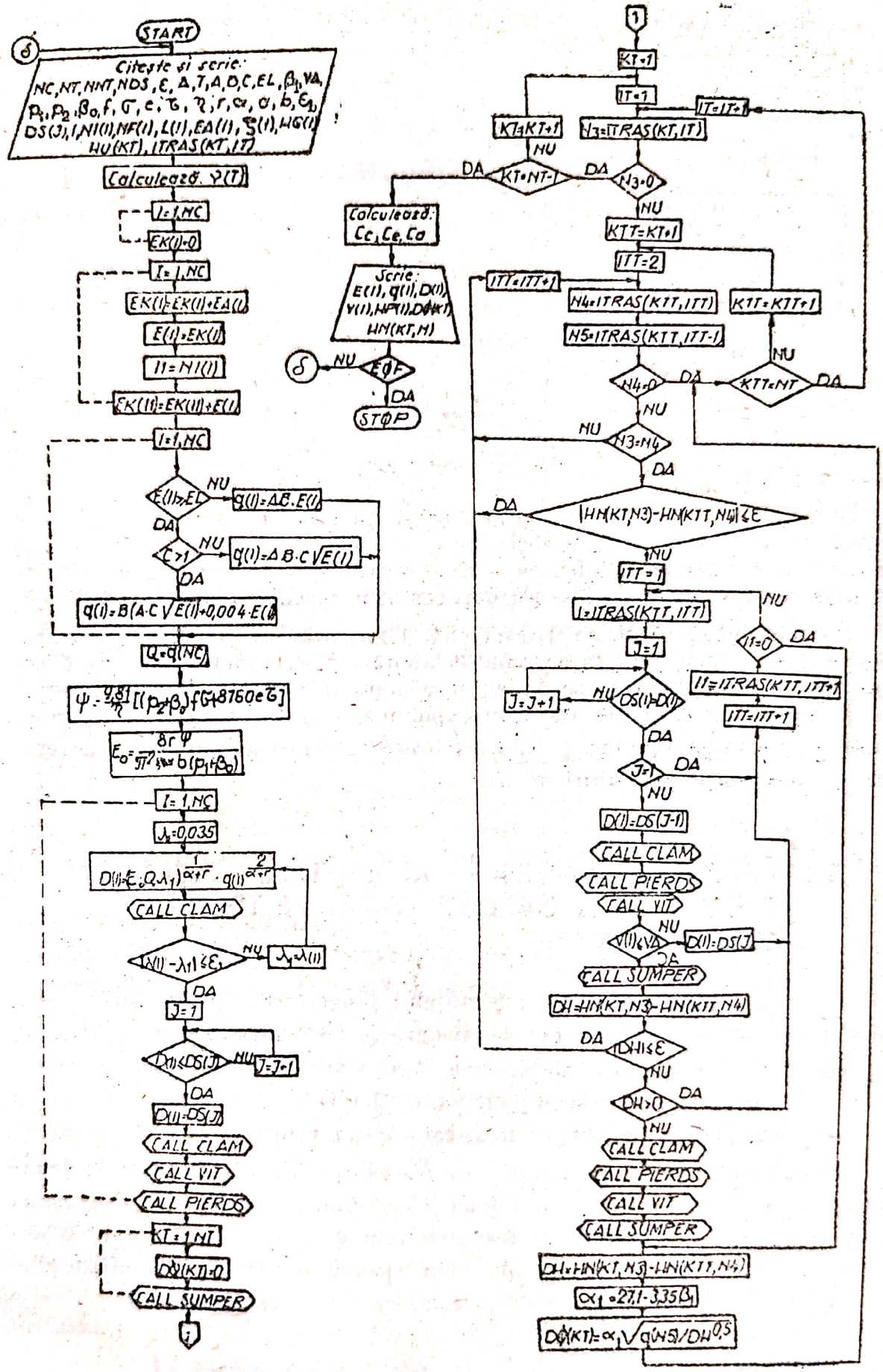


Fig. 5.7. Schema logică a modelului de optimizare a dimensionării rețelelor de alimentare cu apă în clădiri

Scanned with OKEN Scanner

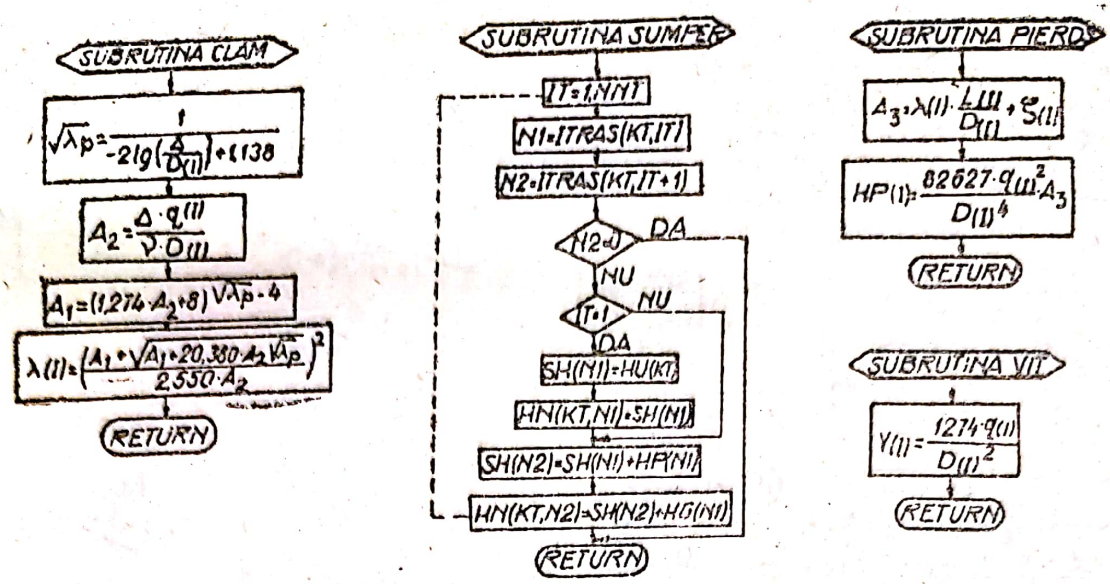


Fig. 5.7. (continuare)

Modelul de calcul expus permite să se țină seama la analiza soluțiilor optime de rețele interioare de distribuție a apei de variația în timp a principalilor parametri economici, cum sînt: costul specific al energiei electrice, duratele de amortizare ale investițiilor, costurile specifice ale materialelor etc.

De menționat că firma ROMTEST Timișoara testează piața materialelor pentru instalații sanitare, avînd colaborări cu mai multe firme austriece care produc conducte și piese de legătură dintr-un material de calitate superioară, polipropilenă de tip Random-Copolymer tratată atît pentru lipire cît și îmbinare prin polifuziune, putînd fi folosite la lucrări noi precum și la reparații și intervenții în clădiri vechi.

5.4. CALCULUL SIMPLIFICAT AL REȚELELOR INELARE DE DISTRIBUȚIE A GAZELOR BAZAT PE TEORIA GRAFURILOR

În scopul îmbunătățirii performanțelor programelor ordinatoare pentru calculul hidraulic laborios al rețelelor inelare de distribuție a gazelor naturale, se poate utiliza un algoritm de schematizare a acestor rețele, bazat pe teoria grafurilor, care permite descompunerea rețelei în subrețele ierarhic dependente prin intermediul unui singur nod, ce se expune în cele ce urmează.

Calculul hidraulic al unei rețele de distribuție a gazelor constă în determinarea debitelor pe tronsoanele rețelei și a presiunilor în nodurile de consum, conducînd, în cazul rețelelor ramificate, la rezolvarea numerică a unui sistem de ecuații format din N-1 ecuații liniare independente de conservare a debitului în noduri și T ecuații neliniare de pierderi de sarcină pe tronsoane:

$$\sum_{\substack{i=1 \\ i\neq j}}^{N} Q_{ij} + q_j = 0 \qquad (j=1, \ldots, N\text{-}1) \qquad (5.136) \qquad X_i - X_j = S_{ij} Q_{ij}^2 \qquad (5.137)$$

Scanned with OKEN Scanner

unde:

$$S_{ij}=r\lambda_{ij}\frac{L_{ij}}{D_{ij}^5} \tag{5.138}$$

r fiind un coeficient ce depinde de natura gazului, de unităţile de măsură adoptate şi de regimul de funcţionare; L_{ij}, D_{ij}, Q_{ij} — lungimea, diametrul şi debitul tronsonului ij; X_i, X_j — presiunile în nodurile i şi j sau pătratul acestor presiuni, după cum conductele funcţionează în regim cu volum constant sau variabil; q_j — debitul consumat în nodul j; λ_{ij} — coeficientul de rezistenţă hidraulică al tronsonului ij.

Pentru reţelele ramificate, acest calcul se conduce în două etape, în prima etapă determinîndu-se debitele cu relaţia (5.136), din aproape în aproape pornind de la extremităţile finale ale ramificaţiilor, iar în a doua etapă calculîndu-se presiunile în noduri cu relaţia (5.137), pornind din nodul sursă.

Pentru calculul reţelelor inelare cu un număr de tronsoane mai mare decît dublul nodurilor acestora se aplică, în general, metoda nodurilor, iar pentru celelalte reţele metoda ciclurilor.

Se consideră îndeplinite ipotezele de conexitate a reţelei şi de unicitate a sursei. Potrivit primei ipoteze graful topologic asociat unei reţele este conex, iar potrivit celei de-a doua ipoteze, cazul unei reţele alimentate de la mai multe surse (fig. 5.8) de presiune dată, se reduce la cazul unei reţele cu o singură sursă prin crearea unui nod fictiv S, căruia i se atribuie o presiune astfel încît să se asigure sensul de curgere în tronsoanele fictive spre nodurile surse reale S_1 şi S_2. De asemenea, se presupune că în aceste tronsoane circulă un debit unitar, iar rezistenţa hidraulică se calculează astfel încît să se asigure presiunea dată în nodurile surse reale.

Se consideră în figura 5.9 o reţea inelară alimentată din mai multe surse (15, 30, 67), după eliminarea ramificaţiilor, şi se expune un algoritm pentru construirea unei reţele ramificate echivalente alimentînd toate nodurile reţelei inelare date, ajungînd la o renumerotare a nodurilor, astfel încît vechiului număr $v(i)$ îi corespunde noul număr i. Algoritmul presupune următoarele etape:

E1) Se marchează sursa $i=0$ şi se pune $j=i$.

E2) Se determină primul tronson nemarcat avînd o extremitate în $v(j)$, iar la cealaltă extremitate un nod nemarcat.

E3) Se marchează tronsonul, se măreşte i cu 1, se marchează cu i cealaltă extremitate a tronsonului şi se revine în E2.

Tronsoanele marcate formează reţeaua ramificată echivalentă reţelei inelare, iar tronsoanele nemarcate, în număr de T-N, denumite legături nu figurează în reţeaua ramificată. Adăugarea succesivă a acestora la reţeaua ramificată formează inelele independente necesare la aplicarea metodei ciclurilor.

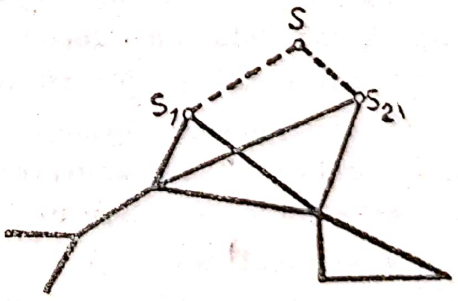


Fig. 5.8. Ipoteza de unicitate a sursei

Scanned with OKEN Scanner

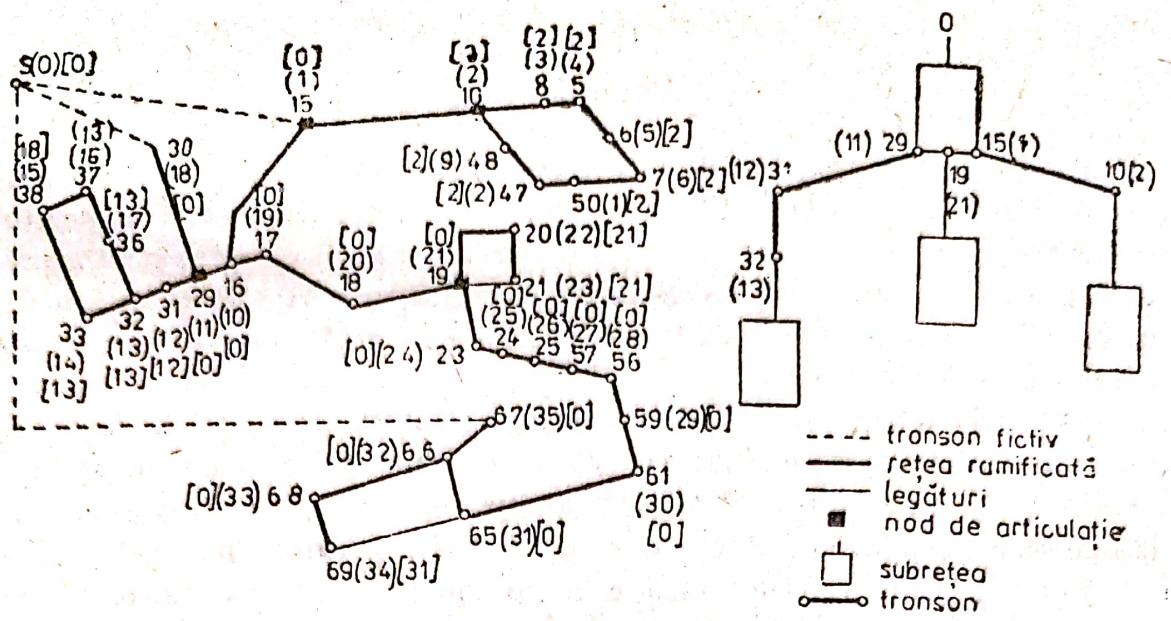


Fig. 5.9. Graful rețelei ramificate echivalente și schema subrețelelor

Se definește funcția $\Lambda(j)$ ca funcția care furnizează mulțimea, eventual vidă, a numerelor i astfel ca tronsoanele ij să fie legături. Se calculează apoi pentru fiecare nod i funcția bază $\beta(i)$ definită prin expresia de recurență:

$$\beta(i)=\min\,[i,\ \Lambda(i),\ \beta(\alpha(i))] \tag{5.139}$$

în care $\alpha(i)$ reprezintă mulțimea, eventual vidă, a nodurilor j adiacente nodului i, care să formeze tronsoane ij.

Dacă pentru cel puțin unul dintre nodurile j adiacente unui nod oarecare i și aparținînd rețelei echivalente este satisfăcută condiția:

$$\beta(j)\geqslant i \tag{5.140}$$

atunci nodul i este un nod de articulație.

Prin scoaterea tronsoanelor avînd extremitatea într-un nod de articulație se separă rețeaua inelară conexă în două subrețele independente.

O subrețea situată între două noduri de articulație joacă rolul unui tronson, iar consumul său se atribuie nodului de articulație avînd în calcul un rol analog nodului final al unei ramificații. Astfel, se poate aplica prima etapă a calculului unei rețele ramificate, iar apoi se determină presiunile în fiecare subrețea începînd cu subrețeaua care conține sursa.

În exemplul din figura 5.9, numerele între parantezele mici au fost atribuite nodurilor prin aplicarea algoritmului descris, iar valorile $\beta(i)$ au fost introduse între paranteze drepte.

Algoritmul prezentat pentru schematizarea rețelelor inelare permite studierea unei modificări locale de structură sau de consum într-o subrețea dată, neținînd seama de influența neglijabilă asupra celorlalte subrețele.

Scanned with OKEN Scanner

5.5. OPTIMIZAREA DIMENSIONĂRII REȚELELOR DE TERMOFICARE

5.5.1. PRINCIPII DE FUNCȚIONARE A UNUI SISTEM DE TERMOFICARE

Un sistem de termoficare permite producerea centralizată a căldurii necesare unui ansamblu de consumatori, care pot fi instalații industriale și instalații centrale de încălzire sau de alimentare cu apă caldă menajeră. El se compune din următoarele elemente principale (fig. 5.10): centralele de producere a căldurii; rețelele de distribuție, constînd din conducte magistrale de ducere și introducere; punctele termice în care se face transferul căldurii în rețelele consumatorilor; instalațiile consumatorilor.

Căldura furnizată de către o centrală de termoficare este transmisă unui agent termic care circulă în rețeaua de distribuție ce face legătura între centrală și consumatori, iar după ce cedează în instalația unui consumator o parte din energia acumulată, agentul de transport se reîntoarce la centrala de termoficare. Se formează astfel un circuit închis, unde fiecare tronson al rețelei de distribuție este compus din două conducte (bitubular) denumite tur și retur.

Ca agent termic se utilizează apa fierbinte sau aburul. Avînd în vedere că aburul permite atingerea de temperaturi mai ridicate, necesare de obicei proceselor tehnologice și că rețelele construite pentru încălzire atît pentru consumatori urbani, cît și industriali utilizează apa ca agent termic, se va face în continuare referire doar la rețelele de termoficare cu apă.

Într-o centrală de termoficare, apa este încălzită de la temperatura de 70°C din conducta de retur pînă la 130...150°C în conducta de tur, în funcție de instalații și modul lor de funcționare. Temperatura minimă este funcție de caracteristicile schimbătoarelor de căldură, iar temperatura maximă depinde de presiunile suportate de instalații, căci apa trebuie menținută sub presiune pentru a fi conservată în stare lichidă. Se admite că, în ciuda pierderilor de căldură din conducte, apa rămîne la o temperatură constantă în toate conductele rețelei de distribuție, ceea ce necesită ca viteza de circulație în conducte să fie superioară unei valori minime date.

Instalația unui consumator reprezintă un element important de conexiune între conducta de tur și cea de retur, conținînd un schimbător de căldură și un sistem de reglaj. În practică se disting două tipuri de instalații, unele cu racordare directă și altele cu racordare indirectă, după cum apa din rețeaua de distribuție circulă în aparatele termice ale consumatorului sau aceasta transmite căldura sa prin intermediul unui schimbător la o rețea secundară. Întrucît problemele rețelei de distribuție sînt de aceeași natură pentru cele două tipuri de sisteme de termoficare, se consideră, pentru simplificare, cazul instalațiilor cu racordare indirectă.

În instalația consumatorului, apa prelevată din conducta de tur a rețelei de distribuție la temperatura t se evacuează din schimbătorul de căldură la temperatura $t-\Delta t$, funcție de t, care este temperatura apei ce circulă prin

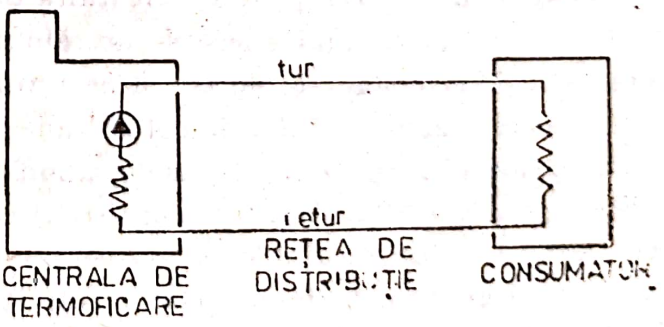


Fig. 5.10 Schema de principiu a unui sistem de termoficare

Scanned with OKEN Scanner

conductele de retur ale rețelei. Sistemul de reglaj permite modificarea debitului de apă ce traversează instalația în funcție de necesarul de căldură și de temperatura t, asigurînd ca diferența de presiune în punctul de conexiune între conducta de tur și retur să fie superioară unei valori minime date.

Circulația apei în circuitul închis format din rețeaua de tur și cea de retur este asigurată de o instalație de pompe de circulație amplasată în centrala de termoficare. Pe de altă parte, pentru reglarea și fixarea presiunii într-un punct al unei conducte a rețelei se prevăd rezervoare de expansiune, iar pentru a se ridica presiunea apei în unele puncte ale rețelei se prevăd stații de pompare intermediare.

În comparație cu regimul de presiune la funcționarea numai a pompelor de circulație, introducerea stațiilor intermediare de pompare pe conducta de tur a rețelei coboară presiunea apei atît în aparatele de încălzire de la centrală, cît și pe porțiunea din conducta de tur dintre cele două instalații de pompare. Introducerea stațiilor intermediare de pompare pe conducta de retur a rețelei coboară, de asemenea, presiunea în schimbătoarele de căldură de la centrală și în porțiunea de conductă dintre cele două stații de pompare. În ambele cazuri presiunea pompelor de circulație se reduce cu presiunea pompelor din stațiile intermediare.

Presiunea apei în toate punctele rețelei trebuie să rămînă între limita minimă necesară pentru evitarea fierberii apei la temperatura de funcționare și limita maximă de nedepășire a rezistențelor mecanice admisibile pentru toate elementele rețelei de termoficare. Aceste condiții asupra presiunii sînt determinante pentru dimensionarea rețelei de distribuție.

Reglarea regimurilor de funcționare a sistemelor de termoficare cu apă se face în funcție de temperatura aerului exterior după următoarele procedee:

— prin varierea temperaturii apei în conducta de tur și de retur, respectiv reglarea calitativă a agentului termic și implicit a debitului de căldură, menținînd constant debitul de apă;

— prin varierea debitului de agent termic, păstrînd constantă temperatura apei, respectiv reglarea cantitativă.

Corectarea reglării centralizate a regimului de funcționare se face prin reglări suplimentare în punctele termice ale industriilor, cartierelor de locuințe, clădirilor și în instalațiile consumatorilor.

5.5.2. ELEMENTE CARACTERISTICE CALCULULUI HIDRAULIC AL REȚELELOR TERMICE

Rețeaua de distribuție este alcătuită din ansamblul tronsoanelor de conducte ce fac legătura între sursele de căldură și consumatori. Fiecare tronson comportă două conducte, de tur și de retur, cu aceleași dimensiuni.

Oricărei rețele termice îi corespunde un graf orientat asociat, compus din arce care pot reprezenta tronsoane, pompe sau armături și vîrfuri ca puncte critice (puncte înalte, stații de pompare), centrale de termoficare, consumatori sau intersecții de conducte.

Topologia unei astfel de rețele poate fi descrisă complet și univoc cu ajutorul matricei de incidență și a matricei ciclurilor construite pentru graful asociat.

Scanned with OKEN Scanner

5.5.2.1. Relația funcțională pierdere de presiune — debit în conductele rețelei

Calculul pierderii de presiune în conductele rețelelor termice se efectuează cu relația de forma generală:

$$\Delta p_{ij}=\frac{8\rho}{\pi^2}\left(\frac{\lambda_{ij}L_{ij}}{D_{ij}}+\zeta_{ij}\right)\frac{G_{ij}^2}{D_{ij}^4} \tag{5.141}$$

în care: L_{ij}, D_{ij}, G_{ij} sînt respectiv lungimea, diametrul și debitul de apă al tronsonului ij; λ_{ij} — coeficientul de rezistență hidraulică longitudinală al tronsonului ij; ζ_{ij} — suma coeficienților de rezistență hidraulică locală a tronsonului ij; ρ — densitatea apei.

În figura 5.11 se prezintă variația presiunii într-o rețea formată dintr-un singur tronson între centrală și consumator, la altitudine constantă, unde: $p_1 - p_2$ este pierderea de presiune în conducta de tur; $p_2 - \bar{p}_2$ — pierderea de presiune în instalația consumatorului; $\bar{p}_2 - \bar{p}_1$ — pierderea de presiune în conducta de retur.

Presupunînd coeficientul de rezistență hidraulică longitudinală constant și neglijînd pierderile locale, se poate folosi pentru calculul aproximativ al pierderii de presiune relația:

$$\Delta p_{ij}=k\frac{L_{ij}}{D_{ij}^5}G_{ij}^2 \tag{5.142}$$

cu $k=8\rho\lambda/\pi^2$.

5.5.2.2. Bazele calculului de dimensionare hidraulică a rețelei

Elementele necesare pentru efectuarea calculului hidraulic al rețelelor de termoficare cu apă sînt următoarele: topologia rețelei; schema de preparare a apei fierbinți în centrală (CET sau CT); puterile termice de calcul ale consumatorilor; parametrii inițiali ai agentului termic.

Prin calculul hidraulic de dimensionare se determină diametrele conductelor rețelei și pierderile de presiune în fiecare tronson, iar pe baza acestui calcul se întocmesc graficele de distribuție a presiunilor în rețea în regim dinamic (de funcționare) și în regim static.

Condițiile extreme de funcționare a rețelei sînt atinse în ziua cea mai friguroasă a anului, consumul de căldură fiind funcție de diferența între temperatura necesară în clădiri și temperatura exterioară. Consumul maxim al unui utilizator, exprimat sub forma puterii termice maxime este calculat în raport cu temperatura exterioară de calcul, considerată în funcție de condițiile climaterice ale zonei în care este amplasat sistemul de termoficare. Determinarea puterii termice, pentru încălzirea clădirilor, se face ținînd seama de volumul și caracteristica termică a acestora.

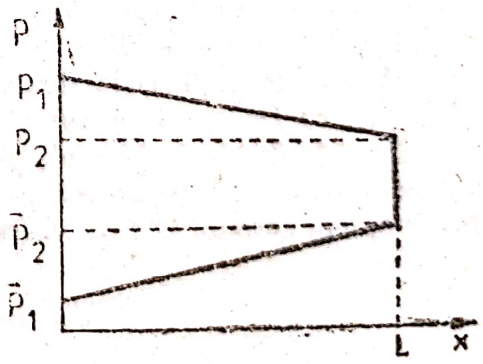


Fig. 5.11. Graficul de presiune al unui tronson de conductă

Necesarul maxim de căldură se exprimă printr-un debit de apă consumat în nodul în care este racordat consumatorul. În ipoteza

Scanned with OKEN Scanner

funcționării sistemului cu diferența maximă de temperatură între rețeaua de tur și cea de retur Δt_{max}, debitul de apă q_j concentrat într-un nod j se exprimă în funcție de puterea termică a consumatorului Q_j, prin relația:

$$q_j = \frac{Q_j}{c\,\Delta t_{max}} \tag{5.143}$$

în care c este căldura specifică a apei la presiune constantă.

Pentru rețelele ramificate, debitele de apă concentrate în noduri determină univoc repartiția debitelor de calcul pe tronsoanele rețelei.

Examinînd diverse tipuri de curbe ale consumului zilnic de căldură, se confirmă că practic toți utilizatorii au un consum maxim fie dimineața, între orele 6 și 10, fie seara, între orele 16 și 20. Se vor considera ca date pentru fiecare consumator atît puterea termică minimă, cît și cea maximă necesară în două momente critice din ziua cea mai friguroasă a anului, ce conduc la debitele de apă corespunzătoare minime și maxime concentrate în fiecare nod j al rețelei, cărora le corespunde debitul mediu q_j.

În calculele de dimensionare optimală a rețelei de distribuție se poate folosi cu suficientă precizie practică, pentru determinarea energiei consumate anual de către o pompă, relația:

$$W = \frac{1}{\eta} G\,\Pi\,T \tag{5.144}$$

în care: η este randamentul general al pompei; T — durata perioadei de încălzire; G — debitul de calcul al conductei considerate, determinat cu ajutorul debitelor medii concentrate în noduri q_j; Π — presiunea de pompare ce asigură funcționarea rețelei sub încărcarea hidraulică medie q_j.

5.5.3. MODEL DE OPTIMIZARE A DIMENSIONĂRII REȚELELOR TERMICE RAMIFICATE BITUBULARE CU APĂ

Avînd în vedere că pe întreaga durată de exploatare a rețelei se consumă energie necesară pentru pomparea apei, dimensionarea optimală a acesteia impune respectarea atît a condițiilor hidraulice de asigurare a debitului de apă și a presiunii în toate punctele de consum, cît și a celor de natură economică, folosindu-se criteriul de optimizare a cheltuielilor anuale de construcție și exploatare minime, pentru o stare staționară a sistemului.

Rețeaua de distribuție fiind ramificată, debitele de calcul pe tronsoane G_{ij} sînt univoc determinate prin debitele concentrate q_j care sînt cunoscute în nodurile rețelei, iar aceste debite de calcul, pentru o încărcare dată a rețelei, au aceeași valoare în rețeaua de tur și cea de retur, doar sensurile fiind schimbate ($\bar{G}_{ij} = -G_{ij}$).

Cunoscînd repartiția debitelor de calcul pe tronsoanele rețelei, se stabilește pentru fiecare tronson ij seria diametrelor comerciale posibil a fi utilizate $D_{k,ij} \in [D_{max,ij}, D_{min,ij}]$, pe baza valorilor limită V_{min} și V_{max} ale vitezei apei ce se impun, respectiv pentru a nu se răci apa prea tare și pentru a evita producerea zgomotelor și vibrațiilor, corespunzător cărora rezultă:

$$D_{max(min),ij} = \sqrt{\frac{4G_{ij}}{\pi V_{min(max),ij}}} \tag{5.145}$$

în care G_{ij} este debitul de calcul al tronsonului ij provenit din încărcarea medie q_j a rețelei.

Scanned with OKEN Scanner

Neglijînd pierderile de presiune locale și admițînd că un tronson ij de lungime L_{ij} al unei rețele alcătuite din T tronsoane este format din s_{ij} sectoare k de diametre comerciale $D_{k,ij}$ și lungimi $X_{k,ij}$, se poate liniariza relația care exprimă diferența de presiune între cele două extremități ale unei conducte:

$$p_i - p_j = \sum_{k=1}^{S_{ij}} \alpha_{k,ij} X_{k,ij} - \Pi_{ij} + \gamma(ZT_j - ZT_i) \tag{5.146}$$

unde:

$$\alpha_{kij} = \frac{8\rho\lambda_{ij} G_{ij} |G_{ij}|}{\pi^2 D_{k,ij}^5} \tag{5.147}$$

în care: Π_{ij} este presiunea activă a stației de pompare intermediare de pe tronsonul ij; ZT_i, ZT_j — cotele geodezice în nodurile i și j; γ — greutatea specifică a apei.

Ținînd seama de expresiile investițiilor în conductele rețelei și în stațiile de pompare intermediare și de costul energiei electrice consumate pentru pompare, funcția obiectiv F_c are forma:

$$F_c = \sum_{ij=1}^{T} \sum_{k=1}^{S_{ij}} c^*_{k,ij} X_{k,ij} + eT \sum_{j=1}^{NS} \frac{1}{\eta_j} q_j (p_j - \bar{p}_j) + \sum_{ij=1}^{NP} [A_{ij} y_{ij} + B_{ij}(r_{ij} + \bar{r}_{ij}) + eT \frac{1}{\eta_{ij}} \cdot (G_{ij}\Pi_{ij} + \bar{G}_{ij}\bar{\Pi}_{ij})] \to \min. \tag{5.148}$$

în care: $c^*_{k,ij}$ este costul specific anual al unui tronson de diametru $D_{k,ij}$, dat de relația (5.114); e — tariful energiei electrice; T — durata perioadei de încălzire; NS — numărul surselor de căldură; NP — numărul stațiilor de pompare intermediare; A_{ij} — costul anual fix pentru o stație de pompare; B_{ij} — costul anual proporțional cu puterea unei stații de pompare; r_{ij}, $\bar{r}_{ij}$ — puterea maximă a unei pompe intermediare care se integrează pe tronsonul ij de tur, respectiv de retur, dacă r_{ij} sau $\bar{r}_{ij}$ rezultă diferit de 0; p_j, $\bar{p}_j$ — presiunea în nodurile j ale rețelei de tur și de retur; G_{ij}, $\bar{G}_{ij}$ — debitele de calcul pe tronsoanele ij ale rețelei de tur și de retur; Π_{ij}, $\bar{\Pi}_{ij}$ presiunile active pe tronsonul ij al rețelei de tur și de retur pe care se poate instala o stație de pompare intermediară, amplasamentul acesteia fiind prin convenție în imediata apropiere a nodului i.

Necunoscutele funcției obiectiv F_c sînt variabilele $X_{k,ij}$, r_{ij}, $\bar{r}_j$, Π_{ij}, $\bar{\Pi}_{ij}$, p_j, $\bar{p}_j$, care trebuie determinate astfel încît să se minimizeze funcția (5.148), în condițiile satisfacerii următoarelor restricții:

— de tip constructiv:

$$\sum_{k=1}^{S_{ij}} X_{k,ij} = L_{ij} \qquad (ij = 1, \ldots, T) \tag{5.149}$$

— de tip funcțional, care se scriu în toate nodurile j sau tronsoanele ij ale rețelei, pentru fiecare din cele trei regimuri de funcționare corespunzătoare puterilor termice necesare, minime, maxime și respectiv medii:

$$p_i - p_j = \sum_{k=1}^{S_{ij}} \alpha_{k,ij} X_{k,ij} - \Pi_{ij} + \gamma(ZT_j - ZT_i) \tag{5.150}$$

$$\bar{p}_i - \bar{p}_j = -\sum_{k=1}^{S_{ij}} \alpha_{k,ij} X_{k,ij} - \bar{\Pi}_{ij} + \gamma(ZT_j - ZT_i) \tag{5.150'}$$

Scanned with OKEN Scanner

$$p_j - \bar{p}_j \geq \delta_j \qquad (5.151)$$

$$r_{ij} \geq G_{ij}\Pi_{ij} \geq 0 \qquad (5.152) \qquad \bar{r}_{ij} \geq \bar{G}_{ij}\bar{\Pi}_{ij} \geq 0 \qquad (5.152')$$

$$r_{ij} \leq M y_{ij} \qquad (5.153) \qquad \bar{r}_{ij} \leq M y_{ij} \qquad (5.153')$$

$$h_j \leq p_j \leq H_j \qquad (5.154) \qquad \bar{h}_j \leq \bar{p}_j \leq \bar{H}_j \qquad (5.154')$$

$$h_j \leq p_j + \Pi_{ij} \leq H_j \qquad (5.155) \qquad \bar{h}_j \leq \bar{p}_j + \bar{\Pi}_{ij} \leq \bar{H}_j \qquad (5.155')$$

$$y_{ij} = \{0, 1\} \qquad (5.156) \qquad X_{k,ij} \geq 0 \qquad (5.157)$$

— de tip hidraulic, care de asemenea se scriu pentru fiecare din cele trei regimuri de funcționare menționate:

$$\sum_{\substack{i=1 \\ i \neq j}}^{N} G_{ij} + q_j = 0 \qquad (j = 1, \ldots, N\text{-}NS) \qquad (5.158)$$

$$\bar{G}_{ij} = -G_{ij} \qquad (ij = 1, \ldots, T) \qquad (5.159)$$

în care: h_j, H_j și $\bar{h}_j$, $\bar{H}_j$ sînt limitele inferioare și superioare ale presiunii în fiecare nod j al rețelei de tur și respectiv de retur; δ_j — diferența minimă de presiune în nodul j între conducta de tur și retur care asigură curgerea debitului q_j prin instalația consumatorului racordat în nodul respectiv.

Restricțiile (5.150) și (5.150′) fac legătura între noile variabile $X_{k,ij}$ și diferența de presiune în nodurile rețelei. Într-o soluție optimă, pentru un tronson dat, cel mult două din aceste variabile vor fi diferite de 0, astfel încît soluția respectivă este realizabilă din punct de vedere tehnic, mai ales dacă lungimea tronsonului este însemnată.

Variabilelor y_{ij} li se atribuie valoarea 0 sau 1, după cum ele descriu decizia de a intercala sau nu o stație de pompare pe tronsonul ij, pe care este posibilă construirea acesteia. Ele sînt legate de puterea maximă a acestor pompe intermediare prin inegalitatea (5.153), unde M reprezintă o constantă cu o valoare suficient de mare.

Relațiile (5.154) și (5.155) limitează intervalul de variație a presiunilor din rețea, fie în noduri, fie la intrarea sau ieșirea din pompele intermediare.

Funcția obiectiv (5.148) și restricțiile (5.149)...(5.159) fiind liniare în raport cu necunoscutele sistemului, soluția optimă se poate determina pe baza metodei programării liniare.

Se observă că prin liniarizarea problemei dimensionării unei rețele termice ramificate se înregistrează o creștere a numărului de variabile. Spre exemplu, pentru o rețea cu 50 noduri și 3 amplasamente posibile pentru stații de pompare intermediare s-a ajuns la un program liniar conținînd circa 550 variabile și tot atîtea necunoscute.

5.6. OPTIMIZAREA DIMENSIONĂRII IZOLAȚIILOR FRIGORIFICE

5.6.1. CONSIDERAȚII PRELIMINARE

Toate instalațiile care au drept scop menținerea temperaturii reci sînt izolate termic, de unde rezultă importanța pe care o primește alegerea corespunzătoare a naturii izolațiilor frigorifice și determinarea grosimii acestora.

Practica actuală a dimensionării izolației încăperilor frigorifice are la bază ecuația de transmitere a căldurii, în care se admite o absorbție de căldură

Scanned with OKEN Scanner

astfel încît nici izolația să nu rezulte prea groasă, deci scumpă, nici consumul de frig să fie foarte mare, iar dimensionarea izolației conductelor frigorifice se efectuează prin respectarea condiției de evitare a fenomenului condensării vaporilor de apă din atmosferă pe suprafața izolației.

Întrucît calculul optimal al izolațiilor frigorifice pe baza criteriului economic al costurilor de investiție și exploatare minime conduce la grosimi de izolații care aplicate în practică devin neoptime la un moment dat după execuție, datorită evoluției prețurilor în timp, s-a conceput un model de dimensionare optimală a acestor izolații cu un grad sporit de generalitate, utilizînd multiple criterii de optimizare dinamice simple și compuse, care reflectă mai bine aspectele cantitative complexe economice și energetice actuale și de perspectivă.

Modelnl de optimizare presupune cunoscute unele date generale și parametrii economico-energetici, date referitoare la încăperi sau conducte și date referitoare la elementele de construcție sau la agentul frigorific transportat.

Conductivitatea termică a materialului izolant λ_{iz} se exprimă sub forma unei dependențe analitice:

$$\lambda_{iz}=b_0+bt_m \tag{5.160}$$

în care: b_0, b sînt constante care depind de natura materialului; t_m — temperatura medie a stratului izolant.

Coeficientul de transmitere a căldurii k se calculează cu relațiile:

— pentru pereții plani ai încăperilor frigorifice:

$$\frac{1}{k}=R=\left(\frac{1}{\alpha_i}+\frac{1}{\alpha_e}+\sum_{m=1}^{NS}\frac{\delta_m}{\lambda_m}\right)+\frac{\delta_{iz}}{\lambda_{iz}} \tag{5.161}$$

— pentru pereții cilindrici ai conductelor frigorifice:

$$\frac{1}{k}=R=\frac{1}{\pi\alpha_i d_i}+\frac{1}{\pi\alpha_e d_p}+\frac{1}{2\pi}\left(\frac{1}{\lambda_c}\ln\frac{d_e}{d_i}+\frac{1}{\lambda_{iz}}\ln\frac{d_{iz}}{d_e}+\frac{1}{\lambda_p}\ln\frac{d_p}{d_{iz}}\right) \tag{5.162}$$

cu: $d_e=d_i+2\delta_c$; $d_{iz}=d_e+2\delta_{iz}$; $d_p=d_{iz}+2\delta_p$, (5.163)

iar temperatura suprafeței interioare a unui perete plan t_{pi} se determină din egalitatea fluxurilor unitare de suprafață:

$$\alpha_i(t_{pi}-t_i)=k(t_e-t_i) \tag{5.164}$$

în care: R este rezistența totală la transfer termic; δ_{iz} — grosimea izolației; δ_m, λ_m — grosimea și conductivitatea termică a stratului m de material component al unui perete plan; δ_c, λ_c — grosimea și conductivitatea termică a peretelui unei conducte; δ_p, λ_p — grosimea și conductivitatea termică a stratului protector al izolației conductei; d_i, d_e — diametrul interior și exterior al unei conducte; t_i — temperatura spațiului răcit; t_e — temperatura exterioară de calcul; α_i, α_e — coeficientul schimbului de căldură la partea interioară, respectiv exterioară a elementului component al spațiului răcit, dat de relația criterială a lui Miheev [90], pentru convecție liberă sau pentru convecție forțată la curgerea turbulentă.

Calculul optimal al grosimii izolațiilor frigorifice presupune minimizarea unor criterii analitice care țin seama de investiția necesară realizării izolației, cheltuielile de exploatare, energia înglobată în izolație, energia consumată pentru menținerea temperaturii scăzute etc., cu respectarea condiției de prevenire a condensării umidității din aer pe suprafața izolației, care în cazul pereților cilindrici are forma:

$$\frac{d_{iz}}{d_e}\ln\frac{d_{iz}}{d_e}\geqslant\frac{2\lambda_{iz}}{\alpha_e d_e}\left(\frac{t_e-t_i}{t_e-t_{pr}}-1\right) \tag{5.165}$$

în care t_{pr} este temperatura punctului de rouă al aerului.

Scanned with OKEN Scanner

5.6.2. CRITERIUL ECONOMIC

Criteriul economic de optimizare a grosimii izolațiilor frigorifice este cel al cheltuielilor echivalente actualizate, care ținînd seama de expresia investiției specifice în izolație I_{iz}, a cheltuielilor anuale pentru întreținerea și repararea izolației C_{ir}, a costului anual al energiei frigorifice pierdute prin suprafața de izolat C_f și cea a ratei r de actualizare a cheltuielilor anuale de funcționare pe perioada timpului normat T de recuperare a izolației:

$$r=\sum_{t=1}^{T}\frac{1}{(1+\beta_0)^t}=\frac{(1+\beta_0)^T-1}{\beta_0(1+\beta_0)^T}, \tag{5.166}$$

presupune minimizarea funcției obiectiv:

$$F_C=I_{iz}+\sum_{t=1}^{T}\frac{C_{ir}+C_f}{(1+\beta_0)^t} \tag{5.167}$$

care pentru pereții plani primește forma (5.168), iar pentru cei cilindrici forma (5.169):

$$F_C=(rp+1)c_{iz}\delta_{iz}+\frac{r\sigma\tau c_f k(t_e-t_i)}{1000}\rightarrow\min \tag{5.168}$$

$$F_C=(rp+1)c_{iz}\delta_{iz}(d_e+\delta_{iz})+\frac{r\sigma\tau c_f k(t_e-t_i)}{1000}\rightarrow\min \tag{5.169}$$

unde: $\beta_0=1/T$ este cota de amortizare; p — cota de reparații, întreținere și revizii pentru izolație; c_{iz} — costul specific de investiție al unui m³ de material izolant; σ — coeficientul pierderilor suplimentare de frig prin armăturile conductei sau datorită regimului nestaționar; τ — numărul de ore necesare furnizării frigului pe perioada unui an; c_f — costul energiei frigorifice.

Valoarea optimă a grosimii izolației se obține prin rezolvarea ecuației $\partial F_c/\partial\delta_{iz}=0$, care pentru pereți plani conduce în final la relația:

$$\delta_{iz}=\frac{r\sigma\tau c_f(t_e-t_i)\lambda_{iz}}{1000(rp+1)c_{iz}}-\lambda_{iz}\left(\frac{1}{\alpha_i}+\frac{1}{\alpha_e}+\sum_{m=1}^{NS}\frac{\delta_m}{\lambda_m}\right), \tag{5.170}$$

iar pentru pereți cilindrici se poate calcula cu metoda aproximațiilor succesive folosind relația (5.169).

5.6.3. CRITERIUL ENERGETIC

Grosimea optimă a izolațiilor, după criteriul energetic, se obține prin minimizarea sumei energiei înglobate în izolație E_{iz} și energia de exploatare E_e necesară menținerii temperaturii scăzute în interiorul spațiului răcit, exprimată analitic prin funcția:

$$F_E=E_{iz}+E_e, \tag{5.171}$$

care pentru pereți plani primește forma (5.172), iar pentru cei cilindrici forma (5.173):

$$F_E=e_{iz}\delta_{iz}+\frac{\sigma\tau T(t_e-t_v)(t_e-t_i)k}{1000\eta_e\eta_f(t_v+273)}\rightarrow\min \tag{5.172}$$

$$F_E=e_{iz}\delta_{iz}(d_e+\delta_{iz})+\frac{\sigma\tau T(t_e-t_i)k}{1000}\rightarrow\min \tag{5.173}$$

Scanned with OKEN Scanner

în care: e_{iz} este energia specifică înglobată într-un m³ de material izolant; t_c, t_v — temperatura de condensare și vaporizare a agentului frigorific; η_e — randamentul mediu de obținere a energiei electrice din energia primară; η_f — randamentul ciclului frigorific real față de un ciclu de referință Carnot parcurs între aceleași temperaturi (fig. 5.12).

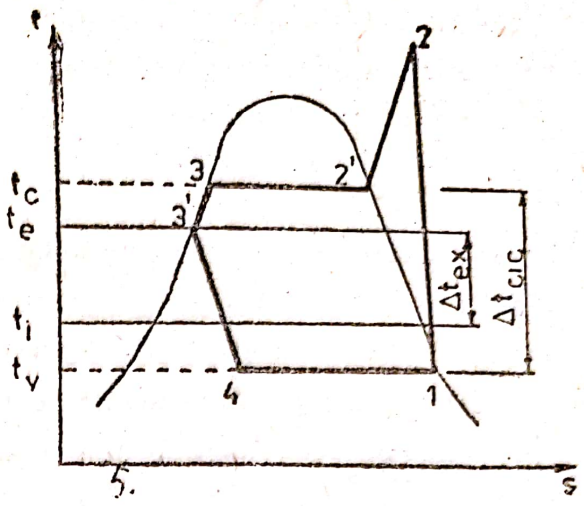


Fig. 5.12. Ciclul termodinamic al unei instalații frigorifice cu comprimare mecanică într-o treaptă

Funcția (5.173) se poate minimiza aplicînd metoda aproximațiilor succesive, iar pentru funcția (5.172) punînd condiția de minim $\partial F_E/\partial\delta_{iz}=0$ se obține, după un șir de transformări algebrice, expresia grosimii optime a izolației pereților plani:

$$\delta_{iz}=\sqrt{\frac{(t_e-t_v)(t_e-t_i)\sigma\tau T\lambda_{iz}}{1000\eta_e\eta_f e_{iz}(t_v+273)}}-\lambda_{iz}\left(\frac{1}{\alpha_i}+\frac{1}{\alpha_e}+\sum_{m=1}^{NS}\frac{\delta_m}{\lambda_m}\right) \qquad (5.174)$$

În urma unor studii comparative, s-a ajuns la concluzia că formula de optimizare (5.174) se poate aplica cu o bună precizie practică și în cazul compresiei în trepte, utilizate frecvent la diferențe mari între temperatura exterioară și cea din spațiul răcit.

5.6.4. CRITERIUL ECONOMICO-ENERGETIC

Determinarea grosimii izolației frigorifice se face pe baza unui calcul care ia în considerare, combinat cele două criterii analizate anterior, reflectînd astfel într-un mod mai obi,ectiv ponderea aspectelor tehnice și a celor energetice pe durata de exploatare a construcției izolate.

Deci, dacă se reprezintă curbele $F_C=f_1(\delta_{iz})$ și $F_E=f_2(\delta_{iz})$ în două plane perpendiculare (fig. 5.13), se observă că punctul M optim absolut ($F_{C,min}$, $F_{E,min}$) se poate reprezenta într-un al treilea plan format din axele F_C și F_E, fiind un punct fictiv întrucît, în general, nu se află în domeniul de existență a legăturii funcționale $F_E=f(F_C)$. Se caută un alt punct N, astfel ca distanța NM să fie minimă

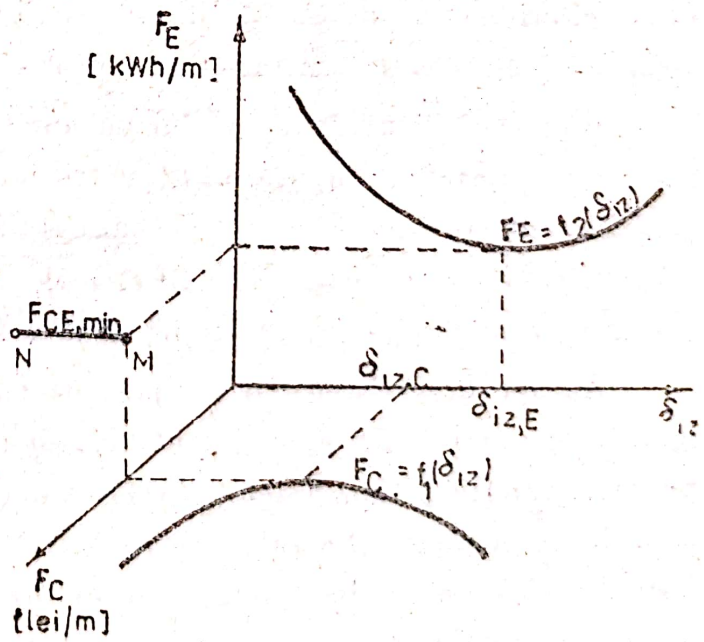


Fig. 5.13. Reprezentarea tridimensională a optimizării bicriteriale a izolației frigorifice

Scanned with OKEN Scanner

și se definește un criteriu complex care să cuprindă pe cele două menționate, sub forma minimizării distanței euclidiene din planul (F_C, F_E):

$$F_{CE}=\sqrt{\psi(F_C-F_{C,min})^2+(1-\psi)(F_E-F_{E,min})^2}\rightarrow\min \qquad (5.175)$$

în care ψ este un coeficient de pondere criterială cu ajutorul căruia, în diverse conjuncturi de prețuri, respectiv de penurie energetică, se poate da preferință unuia sau altuia dintre criteriile componente.

Pentru pereții plani se poate particulariza expresia criteriului complex economico-energetic sub forma analitică:

$$F_{CE}=[\psi(rp+1)c_{iz}+(1-\psi)e_{iz}]\delta_{iz}+\left[\psi r\tau c_f+\frac{(1-\psi)\sigma\tau T(t_e-t_v)}{\eta_e\eta_f(t_v+273)}\right]\cdot$$

$$\cdot\frac{(t_e-t_i)k}{1000}\rightarrow\min \qquad (5.176)$$

Ținînd seama de (5.161) și rezolvînd ecuația $\partial F_{CE}/\partial\delta_{iz}=0$, se obține expresia generală a grosimii optime a izolației frigorifice:

$$\delta_{iz}=\sqrt{\frac{\left[\psi rc_f+\frac{(1-\psi)T(t_e-t_v)}{\eta_e\eta_f(t_v+273)}\right]\frac{(t_e-t_i)\sigma\tau\lambda_{iz}}{1000}}{\psi(rp+1)c_{iz}+(1-\psi)e_{iz}}}-\lambda_{iz}\left(\frac{1}{\alpha_i}+\frac{1}{\alpha_e}+\sum_{m=1}^{NS}\frac{\delta_m}{\lambda_m}\right) \qquad (5.177)$$

Făcînd $\psi=1$ și $\psi=0$ în relația (5.175) și (5.177) se regăsesc cazurile particulare de optimizare corespunzătoare criteriului economic și respectiv energetic, exprimate prin relațiile (5.170) și (5.174) pentru pereții plani și prin relațiile (5.169) și (5.173) pentru pereții cilindrici.

Schemele logice ale modelului de dimensionare optimală a izolației frigorifice se prezintă în figura 5.14, pentru încăperi frigorifice și în figura 5.15 pentru conducte frigorifice, iar pe baza lor s-au elaborat programele ordinatoare DEFRIZOP și respectiv COFRIZOP. Aceste programe cuprind și algoritmul de calcul al grosimii izolației după metoda clasică.

Prin aplicarea criteriului complex de optimizare, pentru o pondere mai mare a criteriului energetic și pentru ponderi egale ale ambelor criterii componente, valorile grosimii izolației rezultă mari, situație ce se poate normaliza admițînd o pondere micșorată pentru criteriul energetic față de cel economic. Valorile mari ale grosimii izolației se datorează în cazul menționat ponderii sporite a energiei necesare menținerii temperaturii în spațiul răcit față de energia înglobată.

Scanned with OKEN Scanner

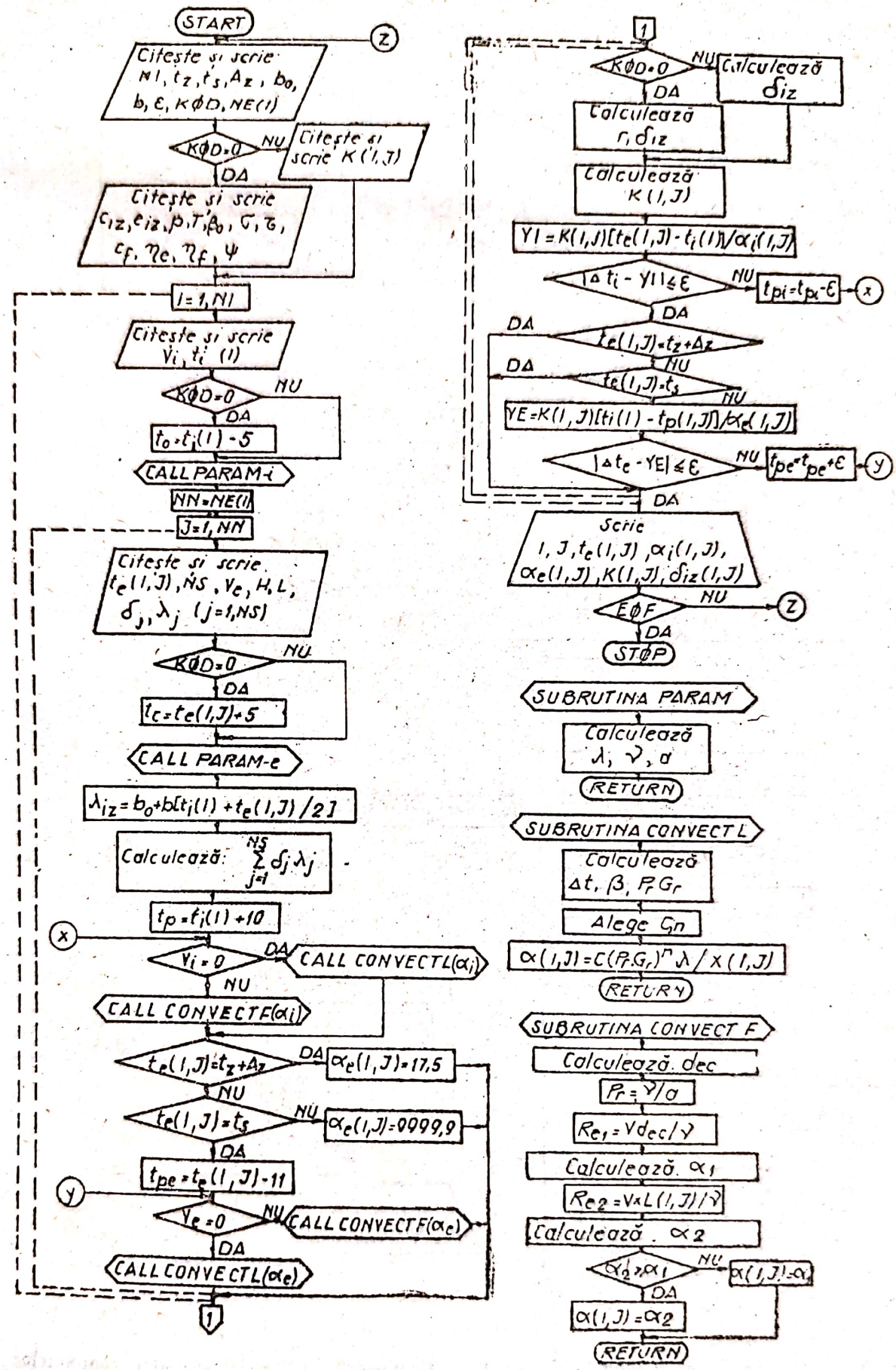


Fig. 5.14. Schema logică a modelului de calcul a grosimii optime a izolației încăperilor frigorifice

Scanned with OKEN Scanner

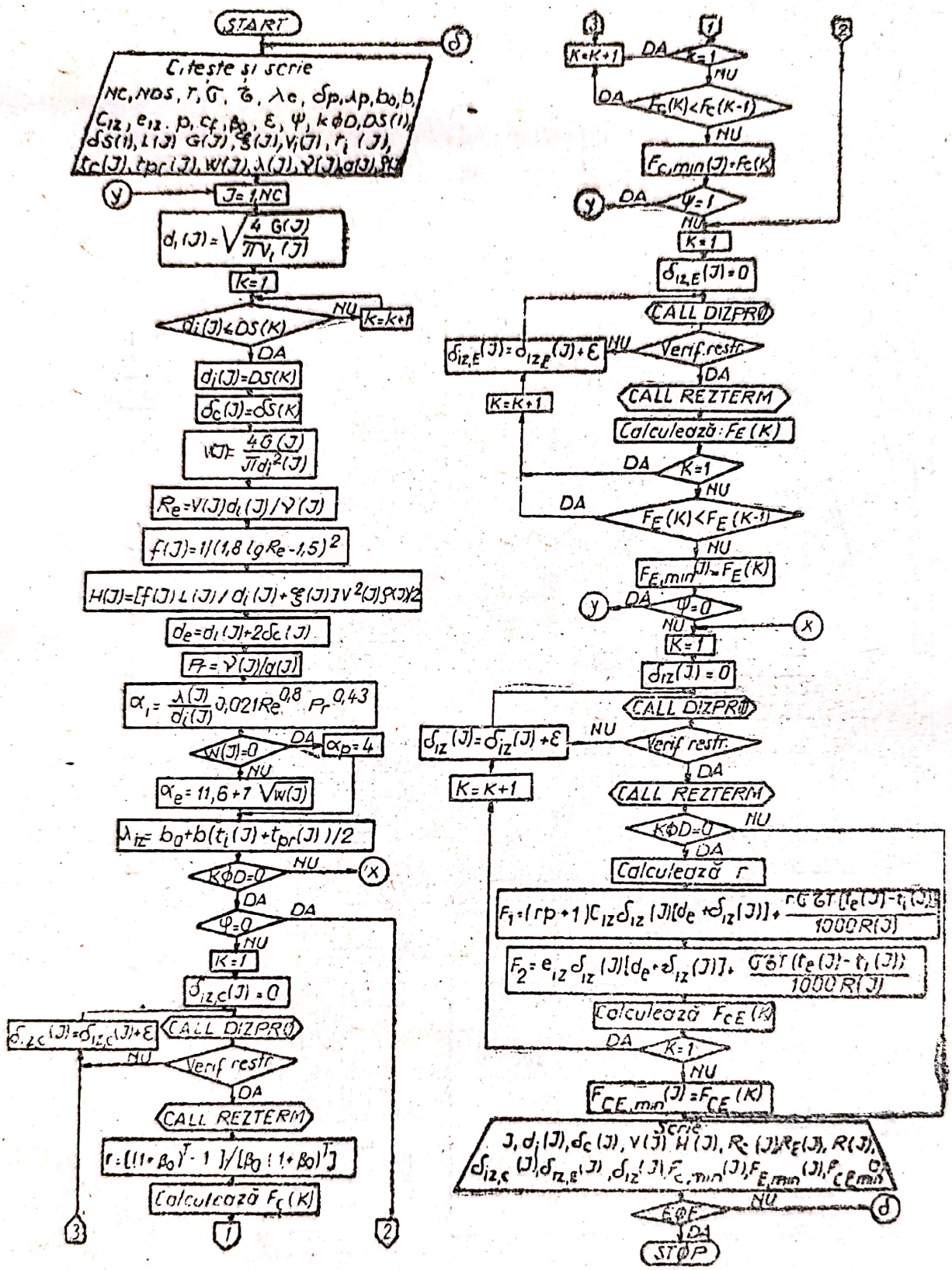


Fig. 5.15. Schema logică a modelului de calcul al grosimii optime a izolației conductelor frigorifice

Scanned with OKEN Scanner

5.7. OPTIMIZAREA ÎNCĂRCĂRII UNOR SISTEME TERMOENERGETICE FUNCȚIONÎND ÎN PARALEL

Se consideră cazul clasic al mai multor cazane de abur funcționînd în paralel pe o conductă colectoare (fig. 5.16).

Caracteristica energetică se exprimă sub forma:

$$C_k = c_{0k} + c_{1k}G_k + c_{2k}G_k^2 \tag{5.178}$$

în care: C_k este consumul de combustibil al cazanului k; G_k — încărcarea (debitul de abur produs) cazanului k.

Funcția obiectiv exprimînd consumul total de combustibil:

$$F(G_k) = \sum_{k=1}^{n}(c_{0k} + c_{1k}G_k + c_{2k}G_k^2) \to \min \tag{5.179}$$

se minimizează în condițiile satisfacerii restricției:

$$\sum_{k=1}^{n} G_k - G_0 = 0 \tag{5.180}$$

Modelul (5.179), (5.180) reprezentînd o problemă de programare neliniară cu restricție de egalitate, se rezolvă aplicînd metoda multiplicatorilor Lagrange. Pentru aceasta se formulează funcția auxiliară:

$$\Gamma = \sum_{k=1}^{n}(c_{0k} + c_{1k}G_k + c_{2k}G_k^2) + \Lambda\left(\sum_{k=1}^{n}(G_k - G_0)\right) \tag{5.181}$$

Extremul căutat se găsește printre soluțiile sistemului:

$$\frac{\partial \Gamma}{\partial G_k} = 0; \qquad \frac{\partial \Gamma}{\partial \Lambda} = 0 \tag{5.182}$$

adică:

$$\frac{\partial C_k}{\partial G_k} + \Lambda = 0; \qquad \sum_{k=1}^{n} G_k - G_0 = 0 \tag{5.183}$$

pentru $k = 1, 2, \ldots, n$.

Prin eliminarea lui Λ rezultă condiția generală de optim, sub forma:

$$\frac{dC_1}{dG_1} = \frac{dC_2}{dG_2} = \ldots = \frac{dC_n}{dG_n} \tag{5.184}$$

combinată cu restricția (5.180).

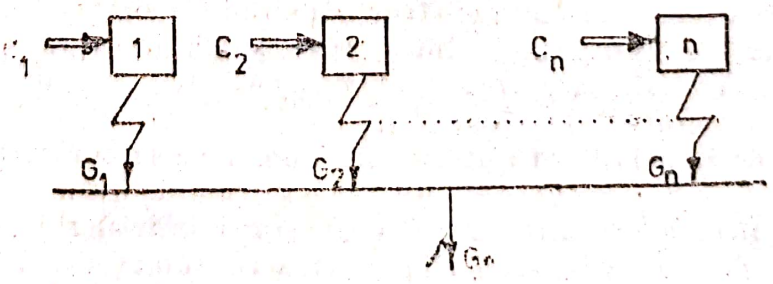


Fig. 5.16. Sdhema funcționării mai multor cazane în paralel

Scanned with OKEN Scanner

Din examinarea relației (5.184) rezultă că optimul se găsește acolo unde tangentele geometrice (a căror înclinare este dată de derivatele dC_k/dG_k) la curbele caracteristicilor energetice sînt paralele între ele. Pentru cazul sistemelor identice, rezultă că optimul căutat este realizat prin încărcarea identică a echipamentelor. Dacă unele echipamente sînt identice ($n_1 < n$), iar restul $n-n_1$ nu, atunci echipamentele identice se încarcă la fel, în timp ce celelalte îndeplinesc condițiile (5.184).

5.8. DETERMINAREA CONFIGURAȚIEI OPTIME A REȚELELOR ELECTRICE

Pentru un sistem format din m surse de energie electrică (centrale electrice) și n consumatori se cunoaște amplasarea surselor și a consumatorilor, puterea disponibilă la fiecare sursă și puterea cerută de consumatori și se pune problema determinării configurației optime din punct de vedere tehnico-economic a rețelei electrice care alimentează consumatorii.

În cazul unor valori mici pentru m și n se selectează cîteva variante, pe baza experienței proiectanților, pentru care se efectuează un calcul tehnico-economic complet, determinîndu-se în final varianta optimă. Pentru dimensiuni mari ale sistemului, alegerea variantelor pentru care se face calculul este dificilă, existînd riscul de a omite varianta optimă. Determinarea variantei optime trebuie să se efectueze pe baza unor metode matematice riguroase, care să țină seama de toate variantele posibile. În mod similar se pun problemele și pentru rețelele termice, de gaze, și de alimentare cu apă.

Drept criteriu de alegere a variantelor optime se poate utiliza minimizarea pierderilor de putere pe ansamblul sistemului, soluționarea completă a problemei expuse presupunînd parcurgerea următoarelor etape:

E1) Verificarea bilanțului puterilor și calculul distanțelor dintre surse și consumatori.

E2) Determinarea riguroasă a schemei sau schemelor optime de alimentare a consumatorilor, pe baza minimizării pierderilor de putere.

E3) Obținerea variantelor pentru care se efectuează calculul tehnico-economic complet, prin corectarea și completarea soluției obținute la etapa E2 (unirea unor trasee de linii care au aproximativ aceeași direcție, adăugarea unor linii din motive de siguranță în funcționare etc.).

E4) Alegerea sau verificarea nivelelor de tensiune ale liniilor, efectuîndu-se corecțiile necesare, cu sau fără revenire la E2.

E5) Efectuarea unui calcul tehnico-economic complet pentru variantele obținute și selectarea soluției optime finale.

În continuare se urmărește soluționarea etapei E2, scop în care se definește următorul model matematic:

a) se iau în considerare numai structurile radiale simple, formate dintr-un număr oarecare de linii radiale între orice sursă i, $i \in E$ (E — mulțimea centralelor electrice) și orice consumator j, $j \in C$ (C — mulțimea consumatorilor), distanța dintre ele fiind L_{ij};

b) sensul de circulație al puterii este numai de la surse către consumatori;

c) pentru fiecare consumator j ($j \in C$), suma puterilor p_{ij} ($i \in E$), transportate pe liniile existente, care alimentează consumatorul j, este egală cu puterea P_j cerută de consumator;

Scanned with OKEN Scanner

d) din punctul de vedere al bilanțului de puteri, pierderile de putere, necunoscute inițial, dar aproximate prin valori medii de calcul, se consideră incluse în puterile cerute de consumatori;

e) pentru fiecare sursă i ($i \in E$) suma puterilor p_{ij} ($j \in C$), de pe liniile existente, care sînt alimentate de la sursa i, trebuie să fie cel mult egală cu puterea disponibilă a sursei P_i;

f) se cere să se determine structura rețelei de alimentare optime, astfel încît pierderile totale de putere pe ansamblul sistemului să fie minime.

Cerințele enunțate la punctele b, c, e se exprimă prin următoarele relații între necunoscutele p_{ij} ale problemei:

$$p_{ij} \geqslant 0, \qquad i \in E; \qquad j \in C \tag{5.185}$$

$$\sum_{j \in E} p_{ij} = P_j, \qquad j \in C \tag{5.186}$$

$$\sum_{j \in C} p_{ij} \leqslant P_i, \qquad i \in E \tag{5.187}$$

Evident, pentru $p_{ij} \neq 0$ există linie între sursa i și consumatorul j, iar pentru $p_{ij} = 0$ nu există.

Bilanțul general al puterii disponibile la surse și al celei cerute de consumatori trebuie să fie de forma:

$$\sum_{i \in E} P_i \geqslant \sum_{j \in C} P_j \tag{5.188}$$

În ipoteza îndeplinirii condiției de egalitate în relația (5.188), relațiile de restricție (5.187) devin:

$$\sum_{j \in C} p_{ij} = P_i, \qquad i \in E \tag{5.189}$$

Condițiile de la punctul f se exprimă printr-o relație de forma:

$$F = \sum_{i \in E} \sum_{j \in C} p_{ij} \to \min \tag{5.190}$$

în care p_{ij} reprezintă pierderile de putere pe linia dintre sursa i și consumatorul j.

În condiții normale, pentru o linie de înaltă sau medie tensiune, pierderile de putere pe rezistența R a liniei sînt dominante, celelalte componente neglijîndu-se într-o primă aproximație (pentru simplificarea relațiilor, se omite scrierea indicilor):

$$p = 3RI^2 = 3\rho \frac{L}{S} I^2 = 3\rho \frac{I}{S} IL = 3\rho \frac{I}{S} L \frac{P}{\sqrt{3}\, U \cos\varphi} = \frac{\sqrt{3}\,\rho j}{U \cos\varphi} PL \tag{5.191}$$

în care: I este intensitatea curentului electric; ρ — rezistivitatea; S — secțiunea; U — tensiunea liniei; j — densitatea de curent; $\cos\varphi$ — factorul de putere.

Presupunînd că durata de utilizare a puterii maxime și factorul de putere au valori apropiate pentru toți consumatorii, rezultă concluzia că expresia $\sqrt{3}\rho j U^{-1} \cos^{-1}\varphi$ este practic constantă, p fiind proporțional cu produsul PL.

În consecință, funcția obiectiv definită de relația (5.190) devine:

$$F = \sum_{i \in E} \sum_{j \in C} L_{ij} P_{ij} \to \min \tag{5.192}$$

Scanned with OKEN Scanner

Problema de optimizare definită de funcţia obiectiv (5.192), de restricţiile (5.186), (5.189) şi de condiţiile de nenegativitate (5.185) este o problemă de programare liniară de forma particulară numită problemă de transport, care se poate soluţiona utilizînd algoritmul Simplex, dar este mai eficientă folosirea unor metode speciale ca metoda Dantzig, metoda elementului minim etc. [18], [54], care solicită un timp de calcul mai redus pe calculator şi mai puţină memorie internă.

5.9. OPTIMIZAREA REPARTIŢIEI SARCINILOR TERMICE ŞI ELECTRICE PE ECHIPAMENTELE UNEI CENTRALE ELECTRICE DE TERMOFICARE ÎN CAZUL UNUI REGIM DAT

Una dintre cele mai eficiente metode de economisire a combustibilului la centrale electrice de termoficare o reprezintă realizarea regimului impus de consumatorul termic şi cel electric printr-o repartiţie optimă a sarcinilor între agregatele centralei (turbine şi staţii de reducere-răcire). Deoarece problema este foarte dificilă în cazuri concrete, se va prezenta doar modelarea sa matematică principială.

Schema generală a unei CET cu conductă colectoare este prezentată în figura 5.17. Ea conţine conducte de abur pentru consumul industrial, conducte de 6 şi 1,2 bar pentru serviciile proprii termice şi posibilităţile de alimentare cu căldură a consumatorului urban.

Consumurile de căldură se pot asigura pe mai multe căi: din prizele turbinelor, de la contrapresiunile altor turbine, de la SRR-uri direct de pe conducta colectoare de abur viu sau prin SRR-uri înseriate. Producţia electrică va fi dependentă de toate aceste debite: direct, prin prizele reglabile sau indirect, prin prizele regenerative care trebuie să încălzească debite diferite de apă de alimentare.

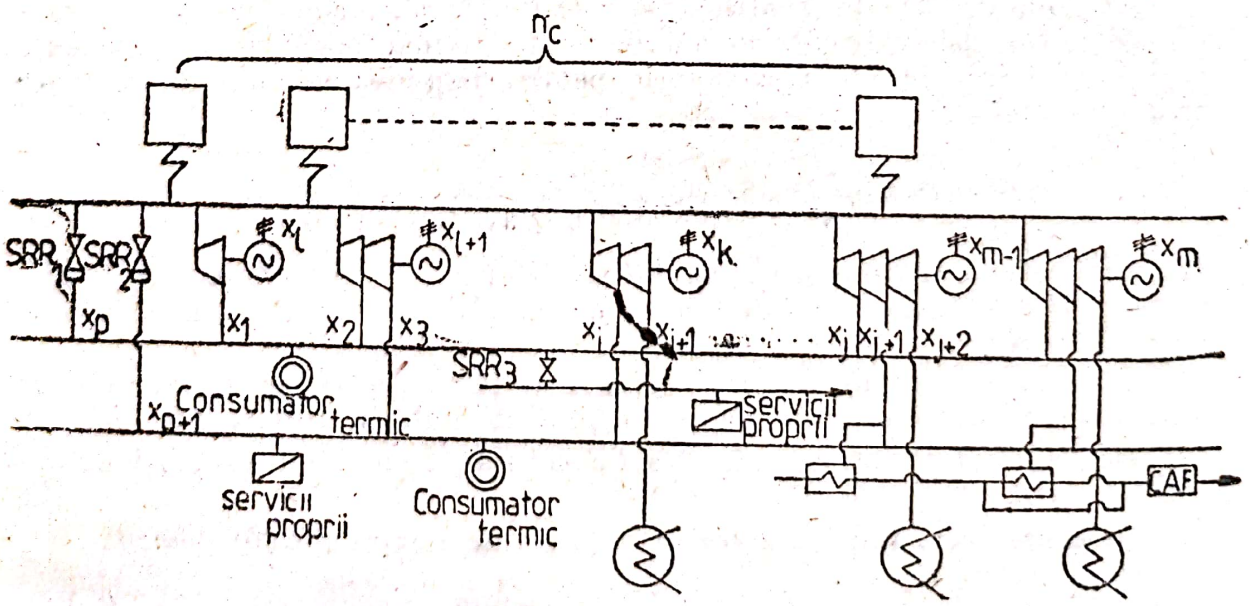


Fig. 5.17. Schema generală a unei centrale electrice de termoficare

Scanned with OKEN Scanner

Pentru modelare, se notează cu x_1, x_2...,x_n variabilele decizionale: debitele de abur la prizele industriale, debitele de abur produse de SRR-uri, căldura dată consumatorului urban din prize, respectiv din cazanele de apă fierbinte, puterile electrice la bornele generatoarelor, debitele de abur la diversele servicii proprii termice.

Funcţia obiectiv o reprezintă consumul total de combustibil pentru întreaga centrală. Este evident că, acest consum depinde de variabilele decizionale definite anterior, adică:

$$F=f(x_1,\ x_2\ ...,x_n) \tag{5.193}$$

Modelul matematic trebuie completat cu diverse restricţii tehnologice, care se pot împărţi în următoarele categorii:

— bilanţuri de masă pe conductele industriale;
— bilanţuri de puteri electrice;
— bilanţuri de căldură la consumatorul urban;
— bilanţuri de masă la serviciile proprii termice şi electrice;
— limitări ale puterii electrice;
— limitări ale debitelor extrase la prize (maxime individuale şi maxime simultane);
— limitări ale debitelor de intrare în turbine şi SRR-uri;
— limitări ale debitelor de abur consumate la serviciile proprii termice.

Toate bilanţurile constituie restricţii de tip egalitate, corespunzîndu-le relaţiile dintre intrări şi ieşiri:

$$\sum_{j=1}^{m} x_j = \sum_{k=1}^{l} x_k \tag{5.194}$$

în timp ce restricţiile de tip limitări sînt inegalităţi:

$$\varphi(x_{i,\ min}) \leqslant \varphi(x_i) \leqslant \varphi(x_{i,\ max}) \tag{5.195}$$

în care: m reprezintă totalitatea variabilelor intrate în conturul de bilanţ; l — totalitatea variabilelor ieşite din contur.

În relaţia (5.195), funcţia φ este dependentă de variabile x_i şi exprimă limitări de încărcare ale echipamentelor centralei. Întrucît restricţiile sînt relaţii liniare, iar funcţia obiectiv conţine caracteristicile energetice ale turbinelor, care sînt tot liniare, programul poate fi tratat ca un program liniar.

5.10. DIMENSIONAREA OPTIMALĂ A UNUI ANSAMBLU DE SCHIMBĂTOARE DE CĂLDURĂ

Schimbătoarele de căldură reprezintă aparate care au drept scop transferul de căldură de la un fluid la altul în procese de încălzire, răcire, fierbere, condensare sau în procese termice în care sînt prezente două sau mai multe fluide cu temperaturi diferite.

În cadrul instalaţiilor tehnologice, aparatele de schimb de căldură ocupă o poziţie particulară, ele putînd funcţiona fie ca organe principale, cînd constituie părţi determinante ale unor procese tehnologice sau ale unor procese exclusiv termice, fie ca organe secundare, introduse în instalaţii din motive de economie de căldură sau de substanţă. Tipul cel mai răspîndit de schimbătoare de căldură îl constituie aparatele recuperatoare.

Scanned with OKEN Scanner

Prin proiectarea optimală a acestor aparate ale instalației tehnologice evident că se obține și o îmbunătățire a performanțelor în faza de exploatare a obiectivelor astfel proiectate.

Se consideră un ansamblu de patru schimbătoare cu parametri cunoscuți, conectate ca în figura 5.18 și se cere o dimensionare a ariilor prin care se realizează transferul de căldură astfel încît temperatura produsului util la ieșirea ultimului schimbător să fie maximă.

Se impune condiția restrictivă ca aria totală de schimb să fie mai mică decît o valoare impusă a.

Pentru ușurarea calculului, dar fără a restrînge generalitatea expunerii, se presupune că agentul termic și produsul util au aceeași capacitate termică, iar debitele sînt aceleași.

În aceste condiții se scriu ecuațiile de bilanț energetic (conservarea energiei termice) în regim staționar de funcționare:

$$m_p c_p (T_j - T_{j-1}) = m_a c_a (t_j - t_{0j}) \tag{5.196}$$

$$kA_j (t_j - T_j) = m_a c_a (t_j - t_{0j}) \tag{5.197}$$

în care: m_p este masa produsului util; m_a — masa agentului termic; c_p — căldura specifică a produsului util; c_a — căldura specifică a agentului termic; k — coeficientul de transfer de căldură; $A_j (j=1, \ldots, 4)$ — ariile prin care se asigură transferul de căldură; $T_j (j=1, \ldots, 4)$ — temperaturile agentului termic la ieșirea fiecărui schimbător; t_j, t_{j0} $(j=1, \ldots, 4)$ — temperaturile de intrare și respectiv ieșire ale agentului termic.

Întrucît se consideră $m_p = m_a = m$; $c_p = c_a = c$ rezultă:

$$T_j - T_{j-1} = t_j - t_{j0} \tag{5.198}$$

$$t_j - t_{j0} = \frac{kA_j}{mc}(t_j - T_j) = \alpha (t_j - T_j) A_j \tag{5.199}$$

în care $\alpha = k/(mc)$ este o constantă.

Prin eliminarea lui t_{0j} între (5.198) și (5.199) se obține relația:

$$T_j = \frac{T_{j-1} + \alpha A_j t_j}{1 + \alpha A_j} \tag{5.200}$$

Modelul matematic al problemei de optimizare enunțate are funcția obiectiv:

$$T_4 \to \max \tag{5.201}$$

și restricțiile:

$$T_j - \frac{T_{j-1} + \alpha A_j t_j}{1 + \alpha A_j} = 0; \qquad a - \sum_{j=1}^{4} A_j = 0 \tag{5.202}$$

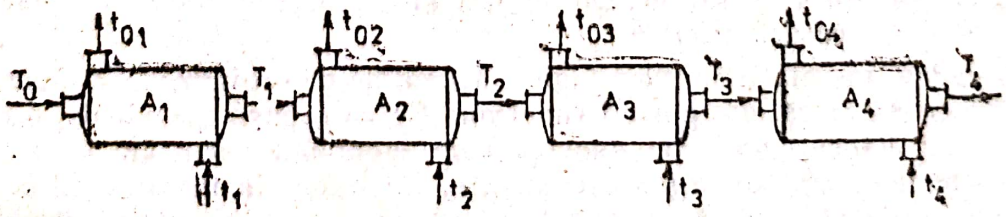


Fig. 5.18. Schema de calcul a unui ansamblu de schimbătoare de căldură

Scanned with OKEN Scanner

Avînd în vedere că toate restricțiile sînt de tip egalitate, rezolvarea cea mai comodă apelează la metoda multiplicatorilor Lagrange.

Se scrie astfel funcția Lagrange corespunzătoare:

$$\Lambda = T_4(T_j,\ A_j) + \lambda_0\left(a - \sum_{j=1}^{4} A_j\right) + \sum_{j=1}^{4} \lambda_j \left(\frac{T_{j-1} + \alpha A_j t_j}{1 + \alpha A_j} - T_j\right) \tag{5.203}$$

Din condițiile necesare de optim în raport cu T_j, A_j, λ_j rezultă următorul sistem de ecuații neliniare:

$$\begin{gathered}
\lambda_j - (1 + \alpha A_j)\lambda_j = 1 \qquad (j = 1,\ 2,\ 3) \\
\lambda_4 = 1 \\
\alpha \lambda_j (t_j - t_{j-1}) = \lambda_0 (1 + A_j)^2 \qquad (j = 1,\ \ldots,\ 4) \\
a - \sum_{j=1}^{4} A_j = 0 \\
\frac{T_{j-1} + \alpha A_j t_j}{1 + \alpha A_j} - T_j = 0
\end{gathered} \tag{5.204}$$

Sistemul (5.204) fiind dificil de rezolvat se preferă o rezolvare numerică de forma (5.203) cu distribuirea calculului la două nivele. La nivelul de coordonare se determină multiplicatorii Lagrange $\lambda_j (j = 1,\ \ldots,\ 4)$ printr-un algoritm de tip gradient după relația iterativă:

$$\lambda_j^{(i+1)} = \lambda_j^i - \left[\frac{\partial \Lambda}{\partial \lambda_j}\right]^{(i)} \quad (j = 1,\ \ldots,\ 4) \tag{5.205}$$

ar la nivelul local se maximizează funcția Λ cu parametrii λ_j furnizați de nivelul coordonator, cu un algoritm de căutare Hooke-Jeeves prin soluționarea succesivă a problemei:

$$\begin{gathered}
T_4(T_j,\ A_j) + \sum_{j=1}^{4} \lambda_j^* \left(\frac{T_{j-1} + \alpha A_j t_j}{1 + \alpha A_j} - T_j\right) \to \max \\
a - \sum_{j=1}^{4} A_j = 0
\end{gathered} \tag{5.206}$$

unde i este indicele etapei de calcul.

5.11. DETERMINAREA MOMENTULUI OPTIM DE ÎNLOCUIRE A CONDUCTELOR AVARIATE ALE REȚELELOR DE TRANSPORT

În cadrul unei rețele de transport (hidraulică, termică, electrică) un pericol potențial ce poate duce la daune, costuri și consumuri energetice suplimentare, precum și la întreruperi în programul normal de funcționare al acesteia sînt avariile ce pot surveni la conducte.

Cauzele principale care conduc la avarierea conductelor se pot clasifica în cîteva categorii generale:

- greșeli de proiectare;
- execuție necorespunzătoare;
- exploatare nerațională;
- calitate și vîrstă a conductelor.

Scanned with OKEN Scanner

Soluționarea problemei avarierii unei conducte în funcțiune poate fi efectuată prin două căi: reparare sau înlocuire, iar luarea unei decizii raționale în acest sens implică o analiză economică de comparare a celor două alternative.

În acest context, se prezintă o metodă generală pentru determinarea momentului optim de înlocuire a conductelor avariate, care nu necesită cunoașterea cauzei avariei ci numai a probabilităților ca avariile să evolueze în timp.

Caracteristica esențială a metodei o reprezintă estimarea realistă a datelor viitoare de avariere, iar analiza economică implicată de aceasta se bazează pe următoarele elemente: numărul estimativ de avarii pe o conductă nouă ca funcție de timp din momentul instalării acesteia; numărul estimativ de avarii pe o conductă existentă; cheltuielile necesare pentru repararea și înlocuirea unei conducte existente cu alta nouă; stabilirea cheltuielilor viitoare la o valoare actualizată.

La înlocuirea unei porțiuni dintr-o conductă avariată cu alta nouă, noua conductă poate avea aceeași calitate ca și cea veche, așteptîndu-se de la ea același istoric al avariilor sau poate avea o calitate superioară, adică să nu apară avarii pe ea. Un mod convenabil de măsurare a avariilor este exprimarea acestora în avarii pe an sau pe metru liniar de conductă. Ecuația de evoluție a avariilor poate fi stabilită pentru o conductă particulară sau pentru o întreagă rețea, în funcție de modul în care este utilizată analiza respectivă.

Se consideră o ecuație de evoluție a avariilor de forma:

$$F(t)=F(t_0)e^{A(t-t_0)} \tag{5.207}$$

în care: t este timpul exprimat în ani; t_0 — timpul de bază al analizei; A — coeficientul de creștere a numărului de avarii; $F(t_0)$ — numărul de avarii pe an.

Cheltuielile de reparație pentru t ani sînt date de relația:

$$c_r=c_aF(t) \tag{5.208}$$

în care c_a reprezintă cheltuielile de reparație a unei avarii.

Valoarea actualizată a cheltuielilor pentru reparații C_r și pentru înlocuirea conductei C_i se determină cu relațiile:

$$C_r=\sum_{t=t_a}^{t_i}\frac{c_r}{(1+\beta_0)^{t-t_*}} \tag{5.209}$$

$$C_i=\frac{c_i}{(1+\beta_0)^{t-t_*}} \tag{5.210}$$

în care: t_* este anul de referință; t_i — anul cînd conducta va fi înlocuită; β_0 — rata de inflație; c_i — costul înlocuirii conductei.

Momentul optim de înlocuire este acel an pentru care cheltuielile totale sînt minime:

$$C_T=\sum_{t=t_*}^{t_i}\frac{c_aF(t_0)e^{A(t-t_0)}}{(1+\beta_0)^{t-t_*}}+\frac{c_i}{(1+\beta_0)^{t_i-t_*}}\to\min \tag{5.211}$$

Prin derivare în raport cu t_i, anulare și rezolvare se obține expresia timpului optim de înlocuire:

$$t_i^0=t_0+\frac{1}{A}\ln\left[\frac{\ln(1+\beta_0)c_i}{F(t_0)c_a}\right] \tag{5.212}$$

Scanned with OKEN Scanner

Se observă că anul de bază t_0 utilizat în dezvoltarea ecuației (5.207) nu influențează asupra soluției optime.

Dacă pentru ecuația de evoluție a avariilor se adoptă forma liniară:

$$F(t)=F(t_0)A(t-t_0) \tag{5.213}$$

atunci rezultă pentru momentul optim de înlocuire a conductei relația:

$$t_i^0=t_0+\frac{\ln(1+\beta_0)c_i}{AF(t_0)c_a} \tag{5.214}$$

Soluțiile optime astfel obținute corespund cazului cînd conducta nouă nu mai poate fi avariată. De aceea se studiază și situația cînd noua conductă poate fi atinsă de avarii, însă începînd cu o valoare inițială mai mică decît cea corespunzătoare coeficientului de creștere caracteristic conductei existente, întrucît altfel nu ar exista nici un motiv pentru înlocuire.

Se consideră că noua conductă este de același tip cu cea existentă și după montare are același istoric al avariilor, iar numărul de avarii crește după același coeficient ca în ecuația (5.207).

Conducta avariată aflată în funcțiune se înlocuiește în anul t_i la un cost c_i, iar odată instalată noua conductă va fi supusă avariilor și se reînlocuiește după t_c ani. Mărimea optimă a ciclului de înlocuire t_c^0 este dată de relația:

$$t_c^0=\frac{1}{A}\ln\left[\frac{\ln(1+\beta_0)c_i}{F(t_0)c_a}\right] \tag{5.215}$$

Este necesar să se definească pentru fiecare ciclu cheltuielile de reparație și de înlocuire exprimate în valori raportate la începerea ciclului. Pentru lungimea optimă a ciclului t_c^0 acestea au respectiv expresiile:

$$C_R=\sum_{t=1}^{t_c^0}\frac{c_r}{(1+\beta_0)^t}=\frac{[(1+\beta)^{t_c^0}-1]c_r}{\beta_0(1+\beta_0)^{t_c^0}} \tag{5.216}$$

$$C_i=\frac{c_i}{(1+\beta_0)^{t_c^0}} \tag{5.217}$$

Prin explicitarea cheltuielilor totale $C_T=C_R+C_i$ se obține relația finală:

$$C_T=\sum_{t=t_*}^{t_i}\frac{c_aF(t_0)e^{A(t-t_0)}}{(1+\beta_0)^{t-t_*}}+\frac{(C_R+c_i)B+c_i}{(1+\beta_0)^{t_i-t_*}} \tag{5.218}$$

unde:

$$B=\sum_{n=1}^{\infty}\frac{1}{(1+\beta_0)^{nt_c^0}}=\frac{1}{(1+\beta_0)^{t_c^0}-1} \tag{5.219}$$

Minimizînd cheltuielile totale se obține, în urma efectuării calculelor pentru momentul optim de înlocuire o expresie similară cu relația (5.212):

$$t_i^0=t_0+\frac{1}{A}\ln\left[\frac{\ln(1+\beta_0)(C_R+c_i)B+c_i}{F(t_0)c_a}\right] \tag{5.220}$$

care conține în cel de-al doilea termen produsul $(C_R+c_i)B$ ce ține seama de ciclurile viitoare de înlocuire și reparații.

Metoda expusă, necesitînd doar cunoașterea unor date referitoare la cheltuielile de reparații și înlocuire, coeficientul de creștere a numărului de avarii și rata de inflație, oferă posibilitatea de calcul a momentului optim de înlocuire a conductelor avariate, asigurîndu-se astfel luarea unor decizii importante în mod corect și cu cheltuieli minime.

Scanned with OKEN Scanner

6

Aplicații numerice și programe de calcul

6.1. CONSIDERAȚII PRELIMINARE

Modelele matematice care rezultă prin aplicarea metodelor numerice și de optimizare în domeniul instalațiilor, pentru sistemele reale se pot soluționa numai cu ajutorul calculatoarelor numerice. Soluționarea manuală este posibilă în exclusivitate pentru cazuri de dimensiuni mai reduse, în condițiile unor ipoteze simplificatoare, dar necesară pentru înțelegerea algoritmilor de calcul și pentru validarea modelelor adoptate și a programelor ordinatoare elaborate, sens în care se prezintă mai multe aplicații numerice.

De asemenea, se dă un minimum de cunoștințe privind limbajul de programare FORTRAN 77, necesare înțelegerii programelor ordinatoare elaborate pe baza modelelor de calcul numeric și de optimizare, cu aplicabilitate generală în domeniul instalațiilor pentru construcții sau pentru rezolvarea unor probleme specializate.

Ideea de bază care a stat la concepția programelor prezentate a fost aceea de a se constitui, alături de o serie de alte programe de calcul specializate existente în biblioteca Catedrei de Instalații a Universității Tehnice din Timișoara, într-un instrument de lucru în activitatea de cercetare, proiectare și didactică. Ele pot fi rulate pe mai multe calculatoare, cu unele modificări neesențiale, dacă ordinatoarele respective dispun de compilatorul asociat limbajului respectiv. Astfel, o altă versiune a acestor programe s-a implementat pe microcalculatoarele IBM PC.

6.2. APLICAȚII NUMERICE

Aplicația 6.2.1

Considerînd variația densității apei ρ în intervalul de temperatură $t = 100...160°C$ din tabelul 6.1, să se determine densitatea apei la 130°C, folosindu-se un polinom Lagrange $\rho(t)$ de gradul 2.

Se selectează ca bază de interpolare punctele: $t_1 = 100$, $t_2 = 120$, $t_3 = 140$ și valorile corespunzătoare ale funcției de densitate.

Tabelul 6.1

t[°C]	100	120	140	160
ρ[kg/m³]	958,38	943,40	926,40	907,50

Scanned with OKEN Scanner

Aplicînd relaţiile (2.4) se obţin coeficienţii Lagrange:

$$w_1 = \frac{(t-t_2)(t-t_3)}{(t_1-t_2)(t_1-t_3)} = \frac{(t-120)(t-140)}{(100-120)(100-140)} = \frac{t^2-260t+16\,800}{800}$$

$$w_2 = \frac{(t-t_1)(t-t_3)}{(t_2-t_1)(t_1-t_3)} = \frac{(t-100)(t-140)}{(120-100)(120-140)} = \frac{t^2-240t-14\,000}{-400}$$

$$w_3 = \frac{(t-t_1)(t-t_2)}{(t_3-t_1)(t_3-t_2)} = \frac{(t-100)(t-120)}{(140-100)(140-120)} = \frac{t^2-220t+12\,000}{800}$$

cu care se determină pe baza relaţiei (2.3) polinomul de interpolare Lagrange:

$$\rho(t) = 958{,}38w_1 + 943{,}40w_2 + 926{,}40w_3 = -0{,}002525t^2 - 0{,}1935t + 1002{,}98$$

şi deci, pentru $t = 130°C$, rezultă:

$$\rho(130) = -0{,}002525 \cdot 130^2 - 0{,}1935 \cdot 130 + 1002{,}98 = 935{,}15 \text{ kg/m}^3.$$

Rezolvarea acestei probleme se poate face folosind programul ordinator APROL 1.

Aplicaţia 6.2.2

Să se determine diferenţele divizate ale funcţiei $f(x)$ date prin tabelul 6.2

Tabelul 6.2

x	0	0,2	0,3	0,4	0,5	0,9
$f(x)$	132,651	148,877	157,464	166,375	195,112	216,000

Aplicînd succesiv formula (2.7), se obţine:

$$[x_1, x_2] = \frac{148{,}877-132{,}651}{0{,}2-0} = 81{,}13$$

$$[x_2, x_3] = \frac{157{,}464-148{,}877}{0{,}3-0{,}2} = 85{,}87$$

.....

$$[x_1, x_2, x_3] = \frac{85{,}87-81{,}13}{0{,}3-0} = 15{,}80$$

.....

Rezultatele calculelor sînt prezentate în tabelul 6.3.

Tabelul 6.3

x	$f(x)$	Diferenţele divizate			
		ordin 1	ordin 2	ordin 3	ordin 4
0	1	2	3	4	5
0	132,651				
		81,13			
0,2	148,877		15,8		
		85,87		1	
0,3	157,464		16,2		0
		89,11		1	
0,4	166,375		16,7		0
		95,79		1	
0,7	195,112		17,3		
		104,44			
0,9	216,000				

Scanned with OKEN Scanner

Aplicația 6.2.3

În cadrul operației de calibrare a unui traductor de debit cu diafragmă, se cere determinarea polinomului Newton de interpolare pătratică $\Delta p(Q)$ a căderii de presiune Δp, cunoscînd din măsurările experimentale valorile acesteia, funcție de debitul Q, date în tabelul 6.4.

Tabelul 6.4

Q[m³/h]	30,28	35,72	38,25	41,08	42,15
Δp[mmCA]	1000	1250	1500	1750	2000

Se calculează în tabelul 6.5 diferențele divizate ale funcției $\Delta p(Q)$ și se selectează ca bază de interpolare punctele: $Q_1=35,72$; $Q_2=38,25$; $Q_3=41,08$

Folosind formula (2.5) se obține expresia polinomului Newton:

$$\Delta p(Q)=-1,9543Q^2+243,3737Q-4949,7837$$

Tabelul 6.5

Q	$\Delta p(Q)$	Diferențele divizate			
		ordin 1	ordin 2	ordin 3	ordin 4
0	1	2	3	4	5
30,28	1000				
		45,9558			
35,72	1250		6,6322		
		98,8142		−0,7950	
38,25	1500		−1,9543		0,5807
		88,3392		6,0983	
41,08	1750		37,2578		
		233,6448			
42,15	2000				

Aplicația 6.2.4

Să se determine funcția binomială de interpolare geometrică a expresiei de variație a energiei specifice înglobate în conductele din azbociment ale rețelelor de distribuție a apei, ale căror valori E_k, în kWh/m, pentru diverse valori ale diametrelor D_k, în m, sînt date în tabelul 6.6.

Din reprezentarea grafică a dependenței $E_k=f(D_k)$, prin extrapolare se determină coeficientul $c_0=40$ al funcției de interpolare (2.8), iar pe baza relațiilor (2.9), (2.10) ținînd seama de calculele efectuate în tabelul 6.6, rezultă

$$\lg c=\frac{25,206\cdot 3,534-10,934\cdot 5,82}{12\cdot 3,534-33,872}=3,024, \text{ sau } c=1060;$$

$$\alpha=\frac{-12\cdot 10,872+5,82\cdot 25,206}{12\cdot 3,534-33,872}=1,90,$$

cu care funcția de interpolare geometrică are expresia:

$$F(D)=40+1060\ D^{1,90}.$$

Pentru rezolvarea acestei probleme se poate utiliza programul ordinator APROGE.

Scanned with OKEN Scanner

Tabelul 6.6

k	D_k	E_k	$\lg(E_k-c_0)$	$\lg D_k$	$(\lg D_k)^2$	$\lg D_k \cdot \lg(E_k-c_0)$
0	1	2	3	4	5	6
1	0,10	50	1,000	−1,000	1,000	−1,000
2	0,15	75	1,544	−0,824	0,679	−1,272
3	0,20	100	1,778	−0,700	0,490	−1,245
4	0,25	125	1,929	−0,602	0,362	−1,161
5	0,30	150	2,041	−0,523	0,274	−1,067
6	0,35	180	2,146	−0,456	0,208	−0,979
7	0,40	200	2,204	−0,398	0,158	−0,877
8	0,45	240	2,301	−0,347	0,120	−0,798
9	0,50	300	2,415	−0,301	0,091	−0,730
10	0,55	360	2,505	−0,260	0,068	−0,651
11	0,60	480	2,643	−0,222	0,049	−0,587
12	0,65	540	2,700	−0,187	0,035	−0,505
	Σ		25,206	−5,820	3,534	−10,872

Aplicaţia 6.2.5

Să se calculeze cu ajutorul formulei Simpson integrala:

$$I=\int_0^1 \frac{dx}{1+x},$$

considerînd numărul diviziunilor intervalului de integrare $2n=10$. Rezultă pasul de integrare:

$$h=\frac{b-a}{2n}=\frac{1-0}{10}=0{,}1$$

iar rezultatele calculelor sînt date în tabelul 6.7.

Cu formula Simpson se obţine:

$$I\simeq\frac{h}{3}[f(0)+f(1)+4\sum_{k=1}^{5}f(x_{2k})+$$

$$+2\sum_{k=2}^{5}f(x_{2k-1})]=\frac{0{,}1}{3}(1+0{,}5+$$

$$4\cdot 3{,}45955+2\cdot 2{,}72818)=0{,}69315$$

La acelaşi rezultat se ajunge folosind programul ordinator SIMPSON.

Tabelul 6.7

k	x_k	$f(x_{2k})$	$f(x_{2k-1})$
0	1	2	3
1	0		1,00000
2	0,1	0,90909	
3	0,2		0,83333
4	0,3	0,76923	
5	0,4		0,71429
6	0,5	0,66667	
	0,6		0,62500
8	0,7	0,58824	
9	0,8		0,55556
10	0,9	0,52632	
11	1,0		0,50000
	Σ	3,45955	2,72818

Aplicaţia 6.2.6

Să se calculeze, aplicînd formula Newton-Côtes cu şapte coordonate ($n=7$), integrala:

$$I=\int_0^1 \frac{dx}{1+x}$$

Scanned with OKEN Scanner

Adoptînd pasul:

$$h=\frac{b-a}{n-1}=\frac{1-0}{7-1}=\frac{1}{6}$$

și ținînd seama de valorile $\bar{w}_k$ ale coeficienților Côtes, majorate de $N=840$ ori, din tabelul 3.2 pentru $n-1=6$, se efectuează calculele corespunzătoare formulei Newton-Côtes în tabelul 6.8.

Tabelul 6.8

k	x_k	$f(x_k)$	$\bar{w}_k$	$\bar{w}_k f(x_k)$
0	1	2	3	4
1	0	1	41	41,000000
2	1/6	6/7	216	185,142857
3	1/3	3/4	27	20,250000
4	1/2	2/3	272	181,333333
5	2/3	3/5	27	16,200000
6	5/6	6/11	216	117,818182
7	1	1/2	41	20,250000
		Σ		581,994372

Ținînd seama că $\sum_{k=1}^{n}\bar{w}_k=N$ rezultă:

$$I=\frac{1}{N}\sum_{k=1}^{7}\bar{w}_k f(x_k)=\frac{1}{840}581,994372=0,6933$$

Valoarea integralei I se poate obține cu ajutorul programului de calcul NEWCOT.

Aplicația 6.2.7

Să se calculeze integrala: $I=\int_0^1\sqrt{1+2x}\,dx$, folosind formula de integrare numerică Gauss-Legendre de ordinul trei ($n=3$).

În acest caz limitele de integrare sînt $a=0$ și $b=1$. Cu ajutorul coordonatelor ξ_k din tabelul 3.3. folosind relația: $x_k=\frac{b+a}{2}+\frac{b-a}{2}\xi_k$ se obțin abscisele punctelor de integrare:

$$x_1=\frac{1}{2}+\frac{1}{2}t_1=\frac{1}{2}-\frac{1}{2}\cdot 0,77459667=0,11270$$

$$x_2=\frac{1}{2}+\frac{1}{2}t_2=\frac{1}{2}+\frac{1}{2}\cdot 0=0,50000$$

$$x_3=\frac{1}{2}+\frac{1}{2}t_3=\frac{1}{2}+\frac{1}{2}\cdot 0,77459667=0,88730$$

Ținînd seama de coeficienții de pondere $\bar{w}_k$ dați în tabelul 3.3. rezultă coeficienții formulei de integrare:

$$C_1=\frac{b-a}{2}W_1=\frac{1}{2}\cdot 0,55555556=0,27778$$

$$C_2=\frac{b-a}{2}W_2=\frac{1}{2}\cdot 0,88888889=0,44444$$

$$C_3=\frac{b-a}{2}W_3=\frac{1}{2}\cdot 0,55555556=0,27778$$

Calculele ulterioare sînt ordonate în tabelul 6.9, obținîndu-se în final valoarea integralei:

$$I=\sum_{k=1}^{3}C_k f(x_k)=1,39870.$$

Scanned with OKEN Scanner

Tabelul 6.9

k	x_k	$f(x_k)$	C_k	$C_k f(x_k)$
0	1	2	3	4
1	0,11270	1,10698	0,27778	0,30747
2	0,50000	1,41421	0,44444	0,62853
3	0,88730	1,66571	0,27778	0,46270
		Σ		1,39870

Aplicația 6.2.8

Să se rezolve folosind metoda Euler, ecuațiile diferențiale:

a) $y'=y$, cu condiția inițială $y(0)=1$ și pentru $x\in[0;\ 0{,}8]$:

b) $y'=\dfrac{y\sqrt{x}-y^2}{\sqrt{x^3}}$, cu condiția inițială $y(1)=2$ și pentru $x\in[1;\ 1{,}8]$.

Tabelele 6.10 și 6.11 conțin valorile obținute pentru soluțiile celor două ecuații, cu relația:

$$y_{k-1}=y_k+hy'(x_k,\ y_k),$$

utilizînd pasul $h=0{,}1$.

Soluțiile celor două ecuații se pot obține folosind programul de calcul EULER.

Aplicația 6.2.9

Să se rezolve, folosind metoda numerică Runge-Kutta, ecuația diferențială:

$y'=(4y-y^2-3)/x$, cu condiția inițială $y(1)=2$ și pentru $x\in[1;\ 1{,}8]$.

Tabelul 6.10

k	x_k	y_k
0	1	2
1	0	1,000000
2	0,1	1,105171
3	0,2	1,221384
4	0,3	1,349907
5	0,4	1,491532
6	0,5	1,650001
7	0,6	1,815693
8	0,7	2,042538
9	0,	2,089871

Tabelul 6.11

k	x_k	y_k
0	1	2
1	1	2,00000
2	1,1	1,80000
3	1,2	1,73681
4	1,3	1,66582
5	1,4	1,61519
6	1,5	1,57889
7	1,6	1,55091
8	1,7	1,52967
9	1,8	1,51998

Scanned with OKEN Scanner

Tabelul 6.12

k	x_k	y_k
0	1	2
1	1	2,00000
2	1,1	2,09518
3	1,2	2,18045
4	1,3	2,25699
5	1,4	2,32468
6	1,5	2,28493
7	1,6	2,43891
8	1,7	2,48599
9	1,8	2,52901

Tabelul 6.12 conține valorile obținute pentru soluția ecuației diferențiale calculată cu relația:

$$y_{k-1}=y_k+\Delta y_k,$$

utilizînd pasul $h=0,1$, unde corecția Δy_k este dată de relația:

$$y_k=1/6(c_1+2c_2+2c_3+c_4),$$

coeficienții c_i avînd expresiile:

$$c_1=hy'(x_k,\ y_k)$$
$$c_2=hy'(x_k+h/2,\ y_k+c_1/2)$$
$$c_3=hy'(x_k+h/2,\ y_k+c_2/2)$$
$$c_4=hy'(x_k+h,\ y_k+c_3).$$

Rezolvarea ecuației diferențiale date se poate efectua utilizînd programul ordinator RUNKUT.

Aplicația 6.2.10

Pentru eliminarea din polinomul:

$$P(x)=x^4-8x^3+5x^2+2x-7$$

a termenului conținînd variabila la puterea a treia, se face schimbarea de variabilă $x=y+2$. Se cere să se determine expresia polinomului transformat $P(y+2)$, precum și derivatele succesive pentru $x=2$ ale polinomului $P(x)$.

Făcînd schimbarea de variabilă $x=y+\xi$, expresia polinomului $P(x)$ devine:

$$P(y+2)=A_0y^4+A_1y^3+A_2y^2+A_3y+A_4$$

în care potrivit lui (2.16) și (2,17): $A_4=P(2)$; $A_{4-i}=P_i(2)$).

Pentru calculul valorilor polinoamelor $P_i(2)$ se utilizează schema lui Horner generalizată:

1	−8	5	2	−7	|2
	2	−12	−14	−24	
1	−6	−7	−12	$-31=P(2)$	
	2	−8	−30		
1	−4	−15	$=42-P_1(2)$		
	2	−4			
1	−2	$-19=P_2(2)$			
	2				
1	$0=P_3(2)$				
$1=P_4(2)$					

de unde se obține: $A_4=-31$, $A_3=-24$, $A_2=-19$, $A_1=0$, $A_0=1$ astfel încît polinomul transformat are expresia:

$$P(y+2)=y^4-19y^2-42y-31$$

Scanned with OKEN Scanner

Derivatele succesive ale polinomului $P(x)$, pentru $x=2$, se calculează cu a doua formulă (2.16):

$$P^{(1)}(2)=1!A_3=-42$$

$$P^{(2)}(2)=2!A_2=-38$$

$$P^{(3)}(2)=3!A_1=0$$

$$P^{(4)}(2)=4!A_0=24$$

Aplicația 6.2.11

Pentru calculul diametrului unei izolații termice la o conductă se folosește relația:

$$\frac{d_{iz}}{d_e}\cdot\ln\frac{d_{iz}}{d_e}=\frac{2\lambda_{iz}(t_f-t_{iz})}{\alpha_e d_e(t_{iz}-t_e)}$$

unde sînt folosite următoarele notații: d_{iz} — diametrul izolației; d_e — diametrul exterior al conductei; λ_{iz} — conductivitatea termică a materialului izolant; α_e — coeficientul de transfer de căldură de la suprafața izolației la mediul ambiant; t_f — temperatura fluxului transportat în conductă; t_{iz} — temperatura pe suprafața stratului de izolație; t_e — temperatura mediului ambiant.

Se consideră o situație practică în care se cunosc următoarele date: $d_e=0{,}219$ m; $\lambda_{iz}=0{,}098$ W/(m°C); $\alpha_e=11{,}5$W/(m²°C); $t_f=350$°C; $t_{iz}=60$°C, $t_e=20$°C și se cere să se determine d_{iz} folosind metoda secantei.

Pentru intervalul de căutare, se consideră $a=0{,}32$, $b=0{,}33$, și $\varepsilon=10^{-3}$. Se notează $x=d_{iz}$ și se construiește funcția:

$$f(x)=\frac{x}{d_e}\ln\frac{x}{d_e}-\frac{2\lambda_{iz}(t_f-t_{iz})}{\alpha_e d_e(t_{iz}-t_e)}$$

Se introduc datele inițiale și se calculează:

$$f(x)=\frac{x}{0{,}219}\ln\frac{x}{0{,}219}-\frac{2\cdot 0{,}098(250-60)}{0{,}219\cdot 11{,}5(60-20)}=\frac{x}{0{,}219}\ln\frac{x}{0{,}219}-0{,}5642$$

$$f(a)=\frac{0{,}32}{0{,}219}\ln\frac{0{,}32}{0{,}219}-0{,}5642=0{,}5541-0{,}5642=-0{,}0101;$$

$$f(b)=\frac{0{,}33}{0{,}219}\ln\frac{0{,}33}{0{,}219}-0{,}5642=0{,}6178-0{,}5642=0{,}0536.$$

Deoarece $f(a)f(b)<0$, se atribuie: $a_1=a=0{,}32$; $b_1=b=0{,}33$ și se calculează:

$$x_1=\frac{a_1f(b_1)-b_1f(a_1)}{f(b_1)-f(a_1)}=\frac{0{,}32\cdot 0{,}0536+0{,}33\cdot 0{,}0101}{0{,}0536+0{,}0101}=0{,}321585,$$

iar apoi:

$$f(x_1)=\frac{0{,}321585}{0{,}219}\ln\frac{0{,}321585}{0{,}219}-0{,}5642=-0{,}0006$$

Deoarece $f(x_1)f(b_1)<0$, se atribuie: $a_2=x_1=0{,}321585$; $b_2=b_1=0{,}33$ și se calculează:

$$x_2=\frac{a_2f(b_2)-b_2f(a_2)}{f(b_2)-f(a_2)}=\frac{0{,}321585\cdot 0{,}536+0{,}33\cdot 0{,}0006}{0{,}0536+0{,}0006}=0{,}321586$$

$$|x_2-x_1|=|0{,}321585-0{,}321586|=10^{-6}$$

Scanned with OKEN Scanner

Întrucît $|x_2-x_1|<\varepsilon$, calculul se oprește, iar soluția problemei este $d_{1,2}=0{,}3215$ m.

Aplicația 6.2.12

Să se determine cu ajutorul metodei Newton, rădăcina reală a ecuației: $f(x)=x^3-x-1=0$, cuprinsă între 1 și 2 cu o eroare $\varepsilon=10^{-3}$.

Deoarece intervalul [a, b]=[1, 2] este relativ mare, se înjumătățește și apoi se aplică metoda Newton:

$$a=1;\ f(a)=-1;\ b=2;\ f(b)=5$$

$$c=\frac{a+b}{2}=\frac{1+2}{2}=1{,}5;\ f(c)=0{,}875$$

Deoarece $f(a)f(c)<0$, se atribuie: $a_1=a=1$ și $b_1=c=1{,}5$.
Pentru obținerea termenilor din șirul de aproximare, se folosește relația:

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}\qquad (k=0,\ 1,\ldots,\ n)$$

Iterația 1: $x_0=1{,}5;\ f(x_0)=0{,}875;\ f'(x_0)=5{,}75$

$$x_1=x_0-f(x_0)/f'(x_0)=1{,}5-0{,}875/5{,}75=1{,}3478$$

$$|x_1-x_0|=|1{,}3478-1{,}5|=1{,}5\cdot 10^{-1}>\varepsilon=10^{-3}$$

Iterația 2: $x_1=1{,}3478;\ f(x_1)=0{,}100566;\ f'(x_1)=4{,}4496$

$$x_2=x_1-f(x_1)/f'(x_1)=1{,}3478-0{,}100566/4{,}4496=1{,}3252$$

$$|x_2-x_1|=|1{,}3252-1{,}3478|=2{,}2\cdot 10^{-2}>\varepsilon=10^{-3}$$

Iterația 3: $x_2=1{,}3252;\ f(x_2)=0{,}0020567;\ f'(x_2)=4{,}2684$

$$x_3=x_2-f(x_2)/f'(x_2)=1{,}3252-0{,}0020567/4{,}2684=1{,}3247$$

$$|x_3-x_2|=|1{,}3247-1{,}3252|=5\cdot 10^{-4}<\varepsilon=10^{-3}$$

Rezultă, deci, soluția $x=x_3=1{,}3247$.

Pentru determinarea rădăcinii reale a ecuației date se poate utiliza programul de calcul ECNEW.

Aplicația 6.2.13

Să se determine inversa matricei:

$$A=\begin{bmatrix} 1 & 2 & 3 \\ -2 & -4 & -5 \\ 3 & 5 & 6 \end{bmatrix}$$

Pornind de la matricea transpusă:

$$A^T=\begin{bmatrix} 1 & -2 & 3 \\ 2 & -4 & 5 \\ 3 & -5 & 6 \end{bmatrix}$$

Scanned with OKEN Scanner

se determină elementele matricei adjuncte A^*:

$$a^*_{11}=\begin{vmatrix}-4 & 5\\ 5 & 6\end{vmatrix};\quad a^*_{12}=-\begin{vmatrix}2 & 5\\ 3 & 6\end{vmatrix};\quad a^*_{13}=\begin{vmatrix}2 & -4\\ 3 & -5\end{vmatrix}$$

$$a^*_{21}=-\begin{vmatrix}-2 & 3\\ -5 & 6\end{vmatrix};\quad a^*_{22}=\begin{vmatrix}2 & 5\\ 3 & 6\end{vmatrix};\quad a^*_{23}=-\begin{vmatrix}1 & -2\\ 3 & -5\end{vmatrix}$$

$$a^*_{31}=\begin{vmatrix}-2 & 3\\ -4 & 5\end{vmatrix};\quad a^*_{32}=-\begin{vmatrix}1 & 3\\ 2 & 5\end{vmatrix};\quad a^*_{33}=\begin{vmatrix}1 & -2\\ 2 & -4\end{vmatrix}$$

cu care se obține:

$$A^*=\begin{bmatrix}-1 & 3 & 2\\ -3 & -3 & -1\\ 2 & 1 & 0\end{bmatrix}$$

Ținînd seama de determinantul matricei A:

$$\dagger A|=\begin{vmatrix}1 & 2 & 3\\ -2 & -4 & -5\\ 3 & 5 & 6\end{vmatrix}=\begin{vmatrix}1 & 2 & 3\\ 0 & 0 & 1\\ 0 & -1 & -3\end{vmatrix}=1\neq 0$$

se determină cu relația (2.19) matricea inversă:

$$A^{-1}=\frac{A^*}{|A|}=\begin{bmatrix}1 & 3 & 2\\ -3 & -3 & -1\\ 2 & 1 & 0\end{bmatrix}$$

Pentru inversarea matricei A se pot utiliza programele ordinatoare INVMAP și INVMAE.

Aplicația 6.2.14

Se consideră că analiza energetică a unui proces de amestec a trei agenți termici, cu trei entalpii diferite, a condus la următorul sistem de ecuații liniare:

$$2x_1+3x_2-x_3=5 \qquad (a)$$

$$4x_1+4x_2-3x_3=3 \qquad (b)$$

$$-2x_1+3x_2-x_3=1 \qquad (c)$$

care se cere să se rezolve cu metoda de eliminare Gauss.

Pentru eliminarea lui x_1 din ecuația (b), se înmulțește ecuația (a) cu multiplicatorul $-4/2$ și se adună rezultatul la ecuația (b), obținîndu-se:

$$-2x_2-x_3=-7 \qquad (b')$$

De asemenea, se înmulțește ecuația (a) cu multiplicatorul $-(-2)/2$ și se adună rezultatul la ecuația (c). Se obține o nouă ecuație:

$$6x_2-2x_3=6 \qquad (c')$$

Scanned with OKEN Scanner

Pentru a elimina necunoscuta x_2 din ecuația (c'), se înmulțește ecuația (b') cu multiplicatorul $-6/(-2)$ și rezultatul obținut se adună la ecuația (c'), rezultînd:

$$-5x_3=-15 \qquad (c'')$$

Cu aceasta, noul sistem de ecuații devine:

$$\begin{aligned} 2x_1+3x_2-\ x_3&=\ \ 5 \qquad (a)\\ -2x_2-\ x_3&=-\ 7 \qquad (b')\\ -5x_3&=-15 \qquad (c'') \end{aligned}$$

iar rezolvarea sa se face folosind algoritmul de substituție înapoi:

$$x_3=b_3/a_{33}=(-15)/(-5)=3$$

$$x_2=(b_2-a_{23}\cdot x_3)/a_{22}=(-7-(-1)\cdot 3)/(-2)=2$$

$$x_1=(b_1-a_{12}\cdot x_2-a_{13}\cdot x_3)/a_{11}=(5-3\cdot 2-(-1)\cdot 3)/2=1$$

Soluția sistemului de ecuații considerat se poate obține cu ajutorul programului de calcul SILGAU.

Aplicația 6.2.15

Să se rezolve prin metoda numerică Cholesky sistemul de ecuații liniare:

$$\begin{aligned} x_1+3x_2-2x_3-\ \ 2x_5&=0{,}5\\ 3x_1+4x_2-5x_3+\ x_4-3x_5&=5{,}4\\ -2x_1-5x_2+3x_3-2x_4+2x_5&=5{,}0\\ x_2-2x_3+5x_4+3x_5&=7{,}5\\ -2x_1-3x_2+2x_3+3x_4+4x_5&=3{,}3 \end{aligned}$$

Se înscriu în coloanele 0...5 ale tabelului 6.13 coeficienții a_{ij} și termenii liberi b_i ai sistemului considerat și se calculează suma algebrică a acestora în coloana 6.

Trecînd succesiv de la o linie la alta, se calculează coeficienții t_{ij} și noii termeni liberi y_i, cu care se completează tabelul 6.13.

Spre exemplu:

$$t_{35}=\frac{1}{t_{33}}(a_{35}-t_{13}t_{15}-t_{23}t_{25})=\frac{1}{0{,}8944i}(2-2\cdot 2-0{,}4472i\cdot 1{,}3416i)=1{,}5653i$$

Pentru verificare se folosește coloana 6 și se calculează apoi valorile necunoscutelor x_i, ce se trec și ele în același tabel, folosind relațiile:

$$x_n=\frac{y_n}{t_{nn}} \qquad x_i=\frac{1}{t_{ii}}\Big(y_i-\sum_{k=i+1}^{n}t_{ik}x_k\Big); \qquad i=n-1,\ \ldots,\ 1$$

Spre exemplu:

$$x_3=\frac{1}{t_{33}}(y_3-t_{35}\cdot x_5-t_{34}\cdot x_4)=\frac{1}{0{,}8944i}(-7{,}7803i-1{,}5652i\cdot 0{,}1998+2{,}0125i\cdot$$

$$\cdot 0{,}8996)=-6{,}8011$$

Scanned with OKEN Scanner

Tabelul 6.13

a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	b_i	Σ
0	1	2	3	4	5	6
1	3	−2	0	−2	0,5	0,5
3	4	−5	1	−3	5,4	5,4
−2	−5	3	−2	2	5,0	1,0
0	1	−2	5	3	7,5	14,5
−2	−3	2	3	4	3,3	7,3
t_{i1}	t_{i2}	t_{i3}	t_{i4}	t_{i5}	y_i	Σ
1	3	−2	0	−2	0,5	0,5
	2,2361i	−0,04472i	−0,4472i	−1,3416i	−1,7471i	−1,747i
		0,8944i	2,0125i	1,5653i	−7,5903i	−3,108i
			3,0414	2,2194	−2,2928	2,9679
				0,8221i	0,1643i	0,9859i
−6,0978	−2,2016	−6,8011	−0,8996	0,1998	x_i	

Aplicația 6.2.16

Să se rezolve aplicînd metoda iterativă Jacobi, sistemul de ecuații liniare:

$$\begin{aligned} 8x_1 + x_2 - x_3 &= 8 \\ 2x_1 + x_2 + 9x_3 &= 12 \\ x_1 - 7x_2 + 2x_3 &= -4 \end{aligned}$$

Se caută pentru fiecare variabilă, cel mai mare coeficient pentru a putea transforma ecuația respectivă în pivot. Se rearanjează ecuațiile astfel:

$$\begin{aligned} 8x_1 + x_2 - x_3 &= 8 \\ x_1 - 7x_2 + 2x_3 &= -4 \\ 2x_1 + x_2 + 9x_3 &= 12 \end{aligned}$$

Se rezolvă fiecare ecuație pentru o singură variabilă:

$$\begin{aligned} x_1 &= 1 - 0{,}125x_2 + 0{,}125x_3 \\ x_2 &= 0{,}571 + 0{,}143x_1 + 0{,}286x_3 \\ x_3 &= 1{,}333 - 0{,}222x_1 - 0{,}111x_2 \end{aligned}$$

Se alege ca soluție inițială: $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$ și rezultă:

$$\begin{aligned} x_1^{(1)} &= 1 - 0{,}125 \cdot 0 + 0{,}125 \cdot 0 = 1 \\ x_2^{(1)} &= 0{,}571 + 0{,}143 \cdot 0 + 0{,}286 \cdot 0 = 0{,}571 \\ x_3^{(1)} &= 1{,}333 - 0{,}222 \cdot 0 - 0{,}111 \cdot 0 = 1{,}333 \end{aligned}$$

Se introduc valorile obținute pentru $x_1^{(1)}$, $x_2^{(1)}$, $x_3^{(1)}$ în membrul drept al sistemului și se obține o nouă soluție:

$$\begin{aligned} x_1^{(2)} &= 1 - 0{,}125 \cdot 0{,}571 + 0{,}125 \cdot 1{,}333 = 1{,}095 \\ x_2^{(2)} &= 0{,}571 + 0{,}143 \cdot 1 + 0{,}286 \cdot 1{,}333 = 1{,}095 \\ x_3^{(2)} &= 1{,}333 - 0{,}222 \cdot 1 - 0{,}111 \cdot 0{,}571 = 1{,}048 \end{aligned}$$

Scanned with OKEN Scanner

Tabelul 6.14

x_i/k	0	1	2	3	4	5	6	7
x_1	0	1,000	1,095	0,995	0,993	1,002	1,001	1,00
x_2	0	0,571	1,095	1,026	0,990	0,998	1,001	1,00
x_3	0	1,333	1,048	0,969	1,000	1,004	1,001	1,00

Procedînd în mod asemănător se obțin rezultatele din tabelul 6.14. Aceste rezultate se pot obține ușor folosind programul ordinator SILJAC.

Aplicația 6.2.17

Să se rezolve prin metoda iterativă Gauss-Seidel sistemul de ecuații liniare:

$$10x_1+ x_2+ x_3=12$$
$$2x_1+10x_2+ x_3=13$$
$$2x_1+ 2x_2+10x_3=14$$

Se transformă sistemul considerat sub o formă comodă pentru procesul de iterație:

$$x_1=1{,}2-0{,}1x_2-0{,}1x_3$$
$$x_2=1{,}3-0{,}2x_1-0{,}1x_3$$
$$x_3=1{,}4-0{,}2x_1-0{,}2x_2$$

Se consideră aproximația inițială $x_1^{(0)}=1{,}2$; $x_2^{(0)}=0$; $x_3^{(0)}=0$ și se aplică succesiv procesul iterativ Gauss-Seidel:

$$x_1^{(1)}=1{,}2-0{,}1\cdot 0-0{,}1\cdot 0=1{,}2$$
$$x_2^{(1)}=1{,}3-0{,}2-0{,}2\cdot 1{,}2-0{,}1\cdot 0=1{,}06$$
$$x_3^{(1)}=1{,}4-0{,}2\cdot 1{,}2-0{,}2\cdot 1{,}06=0{,}948$$
$$x_1^{(2)}=1{,}2-0{,}1\cdot 1{,}06-0{,}1\cdot 0{,}948=0{,}9992$$
$$x_2^{(2)}=1{,}3-0{,}2\cdot 0{,}9992-0{,}1\cdot 0{,}948=1{,}00536$$
$$x_3^{(2)}=1{,}4-0{,}2\cdot 0{,}9992-0{,}2\cdot 1{,}00536=0{,}999098$$

Procedînd în mod asemănător se obțin rezultatele din tabelul 6.15, soluția sistemului fiind: $x_1=1$; $x_2=1$; $x_3=1$. La această soluție se poate ajunge rapid cu ajutorul programului de calcul SILGAS.

Tabelul 6.15

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	1	2	3
0	1,2000	0,0000	0,0000
1	1,2000	1,0600	0,9480
2	0,9992	1,0054	0,9991
3	0,9996	1,0001	1,0001
4	1,0000	1,0000	1,0000
5	1,0000	1,0000	1,0000

Aplicația 6.2.18

Să se determine prin metoda directă, valorile proprii și vectorii proprii ai matricei:

$$A=\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}$$

Scanned with OKEN Scanner

Se scrie ecuația caracteristică a matricei A sub forma:

$$\det(A-\lambda E)=\begin{vmatrix} 2-\lambda & 1 & 1 \\ 1 & 2-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{vmatrix}=0$$

de unde, prin dezvoltarea determinantului se obține ecuația:

$$(\lambda-1)^2(4-\lambda)=0,$$

ale cărei rădăcini: $\lambda_1=\lambda_2=1$, $\lambda_3=4$ sînt valorile proprii ale matricei A.

Introducînd valoarea proprie $\lambda_1=1$ în ecuația matriceală (2.23):

$$\begin{bmatrix} a_{11}-\lambda_k & a_{12} & a_{13} \\ a_{21} & a_{22}-\lambda_k & a_{23} \\ a_{31} & a_{32} & a_{33}-\lambda_k \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}=\begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

aceasta primește forma:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}=\begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

conducînd deci, la sistemul de ecuații:

$$\begin{aligned} x_1+x_2+x_3&=0 \\ x_1+x_2+x_3&=0 \\ x_1+x_2+x_3&=0 \end{aligned}$$

Rangul matricei acestui sistem fiind egal cu 1, este suficient să se rezolve ecuația:

$$x_1+x_2+x_3=0$$

Considerînd $x_1=c_1$ și $x_2=c_2$, se obține: $x_3=-(c_1+c_2)$, unde c_1 și c_2 reprezintă numere care nu sînt simultan nule.

În particular, alegînd la început $c_1=1$, $c_2=0$ și apoi $c_1=0$, $c_2=1$, se obține sistemul fundamental al soluțiilor cele mai simple, compuse din doi vectori proprii ai matricei A, liniar independenți:

$$X^{(1)}=\begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix}; \qquad X^{(2)}=\begin{Bmatrix} 0 \\ 1 \\ -1 \end{Bmatrix}$$

Toți ceilalți vectori proprii ai matricei A, asociate valorii proprii $\lambda_1=1$, sînt combinații liniare a vectorilor de bază $X^{(1)}$ și $X^{(2)}$.

Considerînd $\lambda_3=4$ în ecuația matriceală (2.23), aceasta primește forma:

$$\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}=\begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

conducînd la sistemul de ecuații:

$$\begin{aligned} -2x_1+x_2+x_3&=0 \\ x_1-2x_2+x_3&=0 \\ x_1+x_2-2x_3&=0 \end{aligned}$$

Scanned with OKEN Scanner

Rangul matricei acestui sistem este 2, minorul superior stînga fiind:

$$\Delta=\begin{vmatrix} -2 & 1 \\ 1 & -2 \end{vmatrix}\neq 0$$

Prin urmare, cea de-a treia ecuație se deduce din primele două și se poate reduce sistemul la primele două ecuații:

$$\begin{aligned} -2x_1+\ x_2+x_3&=0\\ x_1-2x_2+x_3&=0, \end{aligned}$$

rezultînd prin aplicarea regulii lui Cramer:

$$\frac{x_1}{\begin{vmatrix} 1 & 1 \\ -2 & 1 \end{vmatrix}}=\frac{x_2}{\begin{vmatrix} -2 & 1 \\ 1 & 1 \end{vmatrix}}=\frac{x_3}{\begin{vmatrix} -2 & 1 \\ 1 & -2 \end{vmatrix}}$$

sau: $x_1/3=x_2/6=x_3/3$, adică: $x_1=x_2=x_3=c$, unde c este o constantă diferită de zero.

Punînd $c=1$ se obține soluția cea mai simplă care exprimă vectorul propriu al matricei A: $X^{(3)T}=\{1\quad 1\quad 1\}$.

Aplicația 6.2.19

Să se determine prin metoda Krîlov, polinomul caracteristic al matricei:

$$A=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

Se alege vectorul inițial $Y^{(0)}=\{1\quad 0\quad 0\quad 0\}$ și apoi se determină coordonatele vectorilor:

$$Y^{(k+1)}=AY^{(k)} \qquad (k=0,\ 1,\ 2,\ 3,\ 4),$$

obținîndu-se:

$$Y^{(1)}=AY^{(0)}=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}\begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix}=\begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}$$

$$Y^{(2)}=AY^{(1)}=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}\begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}=\begin{Bmatrix} 30 \\ 22 \\ 18 \\ 20 \end{Bmatrix}$$

$$Y^{(3)}=AY^{(2)}=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}\begin{Bmatrix} 30 \\ 22 \\ 18 \\ 20 \end{Bmatrix}=\begin{Bmatrix} 208 \\ 178 \\ 192 \\ 242 \end{Bmatrix}$$

$$Y^{(4)}=AY^{(3)}=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}\begin{Bmatrix} 208 \\ 178 \\ 192 \\ 242 \end{Bmatrix}=\begin{Bmatrix} 2108 \\ 1704 \\ 1656 \\ 1992 \end{Bmatrix}$$

Scanned with OKEN Scanner

Se compune sistemul sub forma matriceală:

$$\begin{bmatrix} y_1^{(3)} & y_1^{(2)} & y_1^{(1)} & y_1^{(0)} \\ y_2^{(3)} & y_2^{(2)} & y_2^{(1)} & y_2^{(0)} \\ y_3^{(3)} & y_3^{(2)} & y_3^{(1)} & y_3^{(0)} \\ y_4^{(3)} & y_4^{(2)} & y_4^{(1)} & y_4^{(0)} \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{Bmatrix} = - \begin{Bmatrix} y_1^{(4)} \\ y_2^{(4)} \\ y_3^{(4)} \\ y_4^{(4)} \end{Bmatrix}$$

de unde sub formă detaliată, se obține sistemul de ecuații:

$$\begin{aligned} 2080_1+300_2+\ 0_3+0_4 &= -2108 \\ 1780_1+220_2+20_3 &= -1704 \\ 1920_1+180_2+30_3 &= -1656 \\ 2420_1+200_2+40_3 &= -1992 \end{aligned}$$

prin a cărui rezolvare rezultă: $\theta_1=-4$, $\theta_2=-40$, $\theta_3=-56$, $\theta_4=-20$ și în consecință polinomul caracteristic are expresia:

$$P_4(\lambda)=\lambda^4-4\lambda^3-40\lambda^2-56\lambda-20$$

Pentru determinarea coeficienților acestui polinom se poate utiliza programul ordinator PCKRIL.

Aplicația 6.2.20

Să se determine cu metoda Leverrier, polinomul caracteristic al matricei A definită la exemplul precedent.

Se calculează puterile A^k (k=2, 3, 4) ale matricei A, după cum urmează:

$$A^2=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}=\begin{bmatrix} 30 & 22 & 18 & 20 \\ 22 & 18 & 16 & 18 \\ 18 & 16 & 18 & 22 \\ 20 & 18 & 22 & 30 \end{bmatrix}$$

$$A^3=\begin{bmatrix} 30 & 22 & 18 & 20 \\ 22 & 81 & 16 & 18 \\ 18 & 16 & 18 & 22 \\ 20 & 18 & 22 & 30 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}=\begin{bmatrix} 208 & 178 & 192 & 242 \\ 178 & 148 & 154 & 192 \\ 192 & 154 & 148 & 178 \\ 242 & 192 & 178 & 208 \end{bmatrix}$$

$$A^4=\begin{bmatrix} 208 & 178 & 192 & 242 \\ 178 & 148 & 154 & 192 \\ 192 & 154 & 148 & 178 \\ 242 & 192 & 178 & 208 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}=\begin{bmatrix} 2108 & 1704 & 1656 & 1992 \\ 1704 & 1388 & 1638 & 1656 \\ 1656 & 1638 & 1388 & 1704 \\ 1992 & 1656 & 1704 & 2108 \end{bmatrix}$$

Pentru calcularea sumelor S_k se aplică relația:

$$S_k=\sum_{i=1}^{4} a_{ii}^{(k)}.$$

rezultînd:

$$\begin{aligned} S_1&=1+1+1+1=4 \\ S_2&=30+18+18+30=96 \\ S_3&=208+148+148+208=712 \\ S_4&=2108+1388+1388+2108=6992 \end{aligned}$$

Scanned with OKEN Scanner

Se determină coeficienţii polinomului caracteristic:

$$\theta_K=-\frac{1}{k}\left(S_k+\sum_{i=1}^{k-1}\theta_i S_{k-i}\right);\quad (k=1,\ 2,\ 3,\ 4),$$

obţinîndu-se:

$\theta_1=S_1=-4$

$\theta_2=-\frac{1}{2}(S_2+\theta_1 S_1)=-\frac{1}{2}(96-4\cdot 4)=-40$

$\theta_3=-\frac{1}{3}(S_3+\theta_1 S_2+\theta_2 S_1)=-\frac{1}{3}(712-4\cdot 96-40\cdot 4)=-56$

$\theta_4=-\frac{1}{4}(S_4+\theta_1 S_3+\theta_2 S_2+\theta_3 S_1)=-\frac{1}{4}(6992-4\cdot 712-40\cdot 96-56\cdot 4)=-20$

şi în consecinţă polinomul caracteristic are expresia:

$$P_4(\lambda)=\lambda^4-4\lambda^3-40\lambda^2-56\lambda-20$$

Aplicaţia 6.2.21

Să se determine, cu metoda Newton-Raphson, soluţia pozitivă a sistemului de ecuaţii neliniare:

$$\begin{aligned} x^2+\ y^2+z^2&=1\\ 2x^2+\ y^2-4z&=0\\ 3x^2-4y\ +z^2&=0 \end{aligned}$$

pornind de la aproximaţia iniţială $x^{(0)}=y^{(0)}=z^{(0)}=0{,}5$.

Vectorul funcţie este:

$$F(X)=\begin{Bmatrix} x^2+\ y^2+\ z^2-1\\ 2x^2+\ y^2-4z\\ 3x^2-4y\ +\ z^2 \end{Bmatrix}$$

de unde:

$$F(X_0)=\begin{Bmatrix} 0{,}25+0{,}25+0{,}25-1\\ 0{,}50+0{,}25-2{,}00\\ 0{,}75-2{,}00+0{,}25 \end{Bmatrix}=\begin{Bmatrix} -0{,}25\\ -1{,}25\\ -1{,}00 \end{Bmatrix}$$

Formînd matricea Jacobian:

$$J=\begin{bmatrix} 2x & 2y & 2z\\ 4x & 2y & -4\\ 6x & -4 & 2z \end{bmatrix},$$

rezultă:

$$J_0=\begin{bmatrix} 2\cdot 0{,}5 & 2\cdot 0{,}5 & 2\cdot 0{,}5\\ 4\cdot 0{,}5 & 2\cdot 0{,}5 & -4\\ 6\cdot 0{,}5 & -4 & 2\cdot 0{,}5 \end{bmatrix}=\begin{bmatrix} 1 & 1 & 1\\ 2 & 1 & -4\\ 3 & -4 & 1 \end{bmatrix}$$

şi

$$\det J_0=\begin{vmatrix} 1 & 1 & 1\\ 2 & 1 & -4\\ 3 & -4 & 1 \end{vmatrix}=-40.$$

Scanned with OKEN Scanner

Formînd matricea inversă:

$$J_0^{-1}=1/40\begin{bmatrix}-15 & -5 & -5\\ -14 & -2 & 6\\ -11 & 7 & -1\end{bmatrix}=\begin{bmatrix}3/8 & 1/8 & 1/8\\ 7/20 & 1/20 & -3/20\\ 11/40 & -7/40 & 1/40\end{bmatrix}$$

se determină prima aproximaţie a soluţiei, cu formulele:

$$\Delta X_{k+1}=-J_k^{-1}F(X_k);\qquad X_{k+1}=X_k+\Delta X_{k+1},$$

rezultînd:

$$\Delta X_1=-\begin{bmatrix}3/8 & 1/8 & 1/8\\ 7/20 & 1/20 & -3/20\\ 11/40 & -7/40 & 1/40\end{bmatrix}\begin{Bmatrix}-0{,}25\\ -1{,}25\\ -1{,}00\end{Bmatrix}=\begin{Bmatrix}0{,}375\\ 0\\ -0{,}125\end{Bmatrix}$$

$$X_1=\begin{Bmatrix}0{,}5\\ 0{,}5\\ 0{,}5\end{Bmatrix}+\begin{Bmatrix}0{,}375\\ 0\\ -0{,}125\end{Bmatrix}=\begin{Bmatrix}0{,}875\\ 0{,}500\\ 0{,}375\end{Bmatrix}$$

În mod analog se efectuează şi aproximaţiile ulterioare, iar rezultatele calculului se prezintă în tabelul 6.16.

Oprind procesul iterativ la cea de-a treia aproximaţie, s-a obţinut soluţia X_3 şi reziduul $F(X_3)$:

$$X_3=\begin{Bmatrix}0{,}78521\\ 0{,}49662\\ 0{,}36992\end{Bmatrix};\quad F(X_3)=\begin{Bmatrix}0{,}00001\\ 0{,}00004\\ 0{,}00005\end{Bmatrix}$$

La această soluţie se poate ajunge uşor cu ajutorul programului de calcul SINNER.

Aplicaţia 6.2.22

Să se determine soluţia sistemului de ecuaţii neliniare:

$$x+x^2-2yz=\ \ 0{,}1$$

$$y-y^2+3xz=-0{,}2$$

$$z+z^2+2xy=\ \ 0{,}3.$$

aplicînd metoda gradientului, cu aproximaţia iniţială $x^{(0)}=0$; $y^{(0)}=0$; $z^{(0)}=0$

Tabelul 6.16

k	$\delta_x^{(k)}$	$x^{(k)}$	$\delta_y^{(k)}$	$y^{(k)}$	$\delta z^{(k)}$	$z^{(k)}$
0	1	2	3	4	5	6
0		0,50000		0,50000		0,50000
1	0,37500	0,87500	0	0,50000	−0,12500	0,37500
2	0,08519	0,78981	0,00338	0,49662	0,00507	0,36993
3	−0,00460	0,78521	0	0,49662	−0,00001	0,36992

Scanned with OKEN Scanner

Vectorul funcție și matricea Jacobian au formele:

$$F(X)=\begin{Bmatrix} x+x^2-2yz-0{,}1 \\ y-y^2+3xz+0{,}2 \\ z+z^2+2xy-0{,}3 \end{Bmatrix}, \qquad J=\begin{bmatrix} 1+2x & -2z & -2y \\ 3z & 1-2y & 3x \\ 2y & 2x & 1+2z \end{bmatrix}$$

de unde, cu aproximația inițială, rezultă:

$$F(X_0)=\begin{Bmatrix} -0{,}1 \\ 0{,}2 \\ -0{,}3 \end{Bmatrix}; \qquad J_0=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}=E$$

Se obține soluția în prima aproximație:

$$\alpha_0=\frac{F(X_0)^2}{F(X_0)^2}=1$$

$$X_1=X_0-\alpha_0 EF(X_0)=X_0-1\,EF(X_0)=\begin{Bmatrix} 0{,}1 \\ -0{,}2 \\ 0{,}3 \end{Bmatrix}$$

În mod analog se procedează la cea de-a doua aproximație:

$$F(X_1)=\begin{Bmatrix} 0{,}13 \\ 0{,}05 \\ 0{,}05 \end{Bmatrix}; \qquad J_1=\begin{bmatrix} 1{,}2 & -0{,}6 & 0{,}4 \\ 0{,}9 & 1{,}4 & 0{,}3 \\ -0{,}4 & 0{,}2 & 1{,}6 \end{bmatrix}$$

$$J_1^TF(X_1)=\begin{Bmatrix} 0{,}181 \\ 0{,}002 \\ 0{,}147 \end{Bmatrix}; \qquad J_1J_1^TF(X_1)=\begin{bmatrix} 0{,}2748 \\ 0{,}2098 \\ 0{,}1632 \end{bmatrix}$$

$$\alpha_1=\frac{0{,}13\cdot 0{,}2748+0{,}05\cdot 0{,}2098+0{,}5\cdot 0{,}1632}{0{,}2748^2+0{,}2098^2+0{,}1632^2}=\frac{0{,}054374}{0{,}14619797}=0{,}3719$$

$$x_2=\begin{Bmatrix} 0{,}1 \\ -0{,}2 \\ 0{,}3 \end{Bmatrix}-0{,}3719\begin{Bmatrix} 0{,}181 \\ 0{,}002 \\ 0{,}147 \end{Bmatrix}=\begin{Bmatrix} 0{,}0327 \\ -0{,}2007 \\ 0{,}2453 \end{Bmatrix}$$

Pentru verificare se calculează reziduul:

$$F(X_2)^T=\{0{,}032\ -0{,}017\ 0{,}007\}.$$

Aplicația 6.2.23

Se ilustrează aplicarea modelului matematic expus la pct. 3.1.1 pentru studiul formării gheții și al prevenirii înghețului într-o conductă metalică aeriană caracterizată de parametrii: $G=2{,}0$ m^3/s, $i_c=0{,}015$, $R=0{,}60$ m, $t_e=-10$°C $w=4{,}96$ m/s.

Se determină cu relația (3.5) raza de calcul:

$$R_c=Re^{0{,}615/(wR)^{0{,}7}}=0{,}60e^{0{,}615/(4{,}96\cdot 0{,}60)^{0{,}7}}=0{,}80 \text{ m}.$$

iar apoi pentru raportul $R/R_c=0{,}6/0{,}8=0{,}75$, din (3.20) se obține raza relativă admisibilă $r_{*ad}=0{,}705R/R_c=0{,}705\cdot 0{,}75=0{,}528$.

Scanned with OKEN Scanner

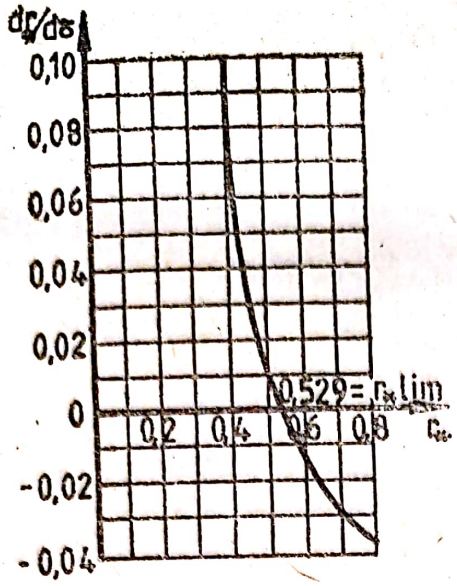


Fig. 6.1. Graficul funcției $dr_*/d\tau$

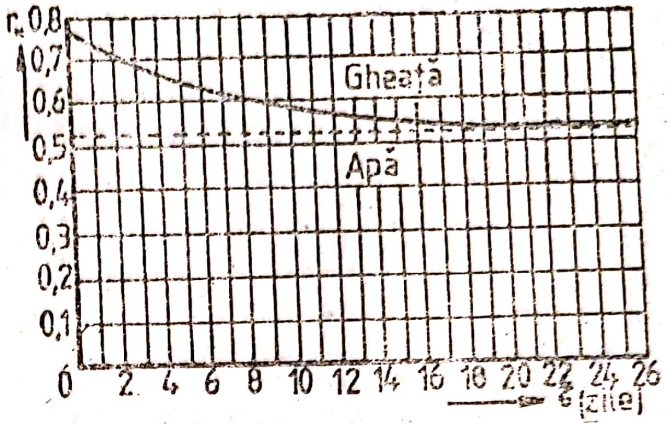


Fig. 6.2. Diagrama de variație a grosimii gheții în timp

În figura 6.1 se trasează graficul funcției $dr_*/d\tau = f(r_*)$ după ecuația diferențială (3.17):

$$\frac{dr_*}{d\tau} = 69{,}375 \cdot 10^{-4} \frac{1}{r_*} + 30{,}714 \cdot 10^{5} \frac{1}{r_*^{6{,}33}} + 10{,}312 \cdot 10^{-3} \frac{1}{r_* \ln r_*}$$

de unde se obține: $r_{*lim} = 0{,}529$.

Pentru rezolvarea ecuației diferențiale (3.17) se poate aplica, în scopul creșterii operativității și a preciziei calculelor, și metoda numerică Runge-Kutta implementată în programul ordinator EVGHECA, iar pe baza rezultatelor obținute se trasează în figura 6.2 diagrama de variație a grosimii gheții în timpul formării acesteia, care tinde asimptotic la orizontala $r_{*lim} = 0{,}529$, rezultînd $\tau_{lim} = 26$ zile.

Întrucît este satisfăcută condiția (3.21): $r_{*lim} = 0{,}529 > r_{*ad} = 0{,}528$, rezultă că nu este necesară izolarea termică a conductei în vederea prevenirii înghețului.

Aplicația 6.2.24

Se aplică modelul matematic expus la pct. 3.2.3 pentru analiza rețelei de distribuție a apei din figura 6.3 realizată din fontă, pentru care se cunosc:

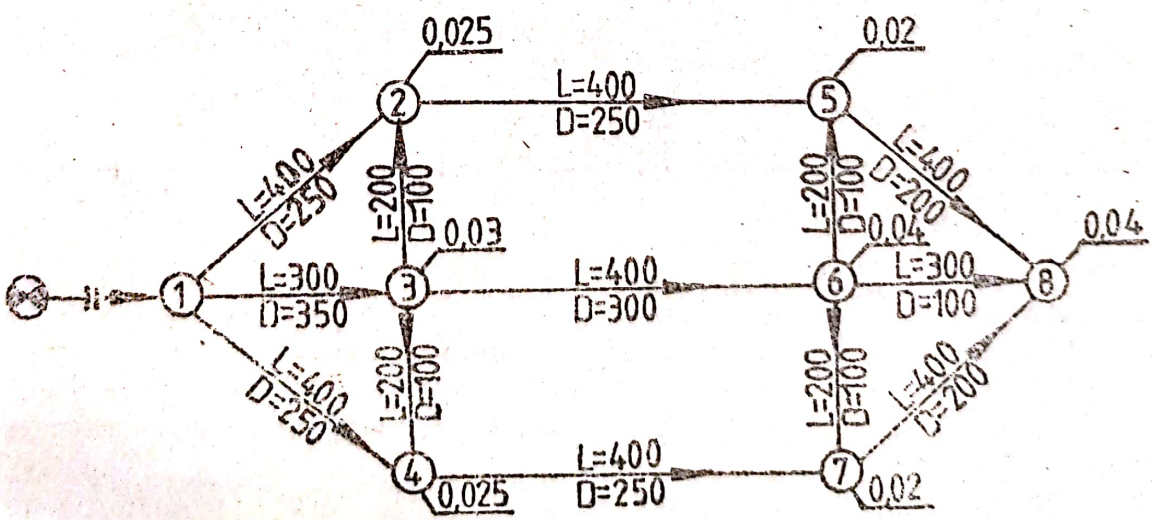


Fig. 6.3. Schema rețelei de distribuție analizate

Scanned with OKEN Scanner

lungimile L_{ij}, în m, diametrele D_{ij}, în mm, ale tronsoanelor, debitul de alimentare $Q=0{,}2$ m³/s, cota piezometrică în punctul obligat $Z_8=100$ m și exponentul $\beta=1{,}93$.

În tabelul 6.17 se prezintă valorile rezistențelor hidraulidce S_{ij} calculate cu relația (5.38), iar cu notația $x=1/\beta$, ecuațiile (3.28) au expresiile:

$$587^{-x}(Z_2-Z_1)^x+79{,}66^{-x}(Z_3-Z_1)^x+587^{-x}(Z_4-Z_1)^x+0{,}2=0$$
$$587^{x}(Z_1-Z_2)^x+31414{,}7^{-x}(Z_3-Z_2)^x+587^{-x}(Z_5-Z_2)^x-0{,}02=0$$
$$79{,}67^{-x}(Z_1-Z_3)^x+31414{,}7^{-x}(Z_2-Z_3)^x+31414{,}7^{-x}(Z_4-Z_3)^x+231{,}5^{-x}(Z_6-Z_3)^x-0{,}03=0$$
$$587^{-x}(Z_1-Z_4)^x+31414{,}7^{-x}(Z_3-Z_4)^x+587^{-x}(Z_4-Z_7)^x-0{,}025=0$$
$$587^{-x}(Z_2-Z_5)^x+31414{,}7^{-x}(Z_6-Z_5)^x+1830^{-x}(Z_8-Z_5)^x-0{,}02=0$$
$$231{,}5^{-x}(Z_3-Z_6)^x+31414{,}7^{-x}(Z_5-Z_6)^x+31414{,}7^{-x}(Z_7-Z_6)^x+47122^{-x}(Z_8-Z_6)^x-0{,}04=0$$
$$587^{-x}(Z_4-Z_7)^x+31414{,}7^{-x}(Z_6-Z_7)^x+1830^{-x}(Z_8-Z_7)^x-0{,}02=0$$

în care apar ca necunoscute cotele piezometrice nodale Z_j ($j=1, \ldots, 7$).

Aplicînd pentru rezolvarea numerică a sistemului de ecuații neliniare astfel format, algoritmul Newton-Raphson implementat în programul ordinator SINNER, s-au determinat cotele piezometrice în nodurile rețelei Z_j ($j=1, \ldots, 7$) ale căror valori sînt prezentate în tabelul 6.18. Cu aceste valori, folosind relația (3.27), s-au calculat debitele de tranzit în conductele rețelei înscrise în tabelul 6.17. La același rezultat s-a ajuns aplicînd și modelul de analiză nodală în formulare variațională implementat în cadrul programului ANOREV.

Alegînd o aproximație inițială a soluției, în limitele de mărime a cotei piezometrice a nodului 8, algoritmul a condus la convergență după 10 iterații.

Tabelul 6.17

$i-j$	S_{ij}[s²/m⁵]	Q_{ij}[m³/s]
0	1	2
1—2	587,00	0,05256
2—5	587,00	0,03247
5—8	1830,00	0,01741
1—3	79,67	0,09501
3—6	231,50	0,05504
6—8	47122,00	0,00515
1—4	587,00	0,05256
4—7	587,00	0,03247
7—8	1830,00	0,01741
3—2	31414,70	0,00501
3—4	31414,70	0,00501
6—5	31414,70	0,00485
6—7	31414,70	0,00485

Tabelul 6.18

j	Z_j [m]
0	1
1	103,53
2	101,53
3	102,68
4	101,53
5	100,74
6	101,82
7	100,74
8	100,00

Aplicația 6.2.25

Se ilustrează rezultatele aplicării metodei elementelor finite la analiza cîmpurilor termice pentru două structuri diferite de izolație destinate conductelor de termoficare pentru transportul căldurii la mari distanțe. În figura 6.4 se prezintă schematic structura celor două tipuri de izolație termică:

— izolație compactă formată dintr-un singur strat de material izolant de bază și un strat protector (fig. 6.4, a);

— izolație cu inele de aer (fig. 6.4, *b*).

Scanned with OKEN Scanner

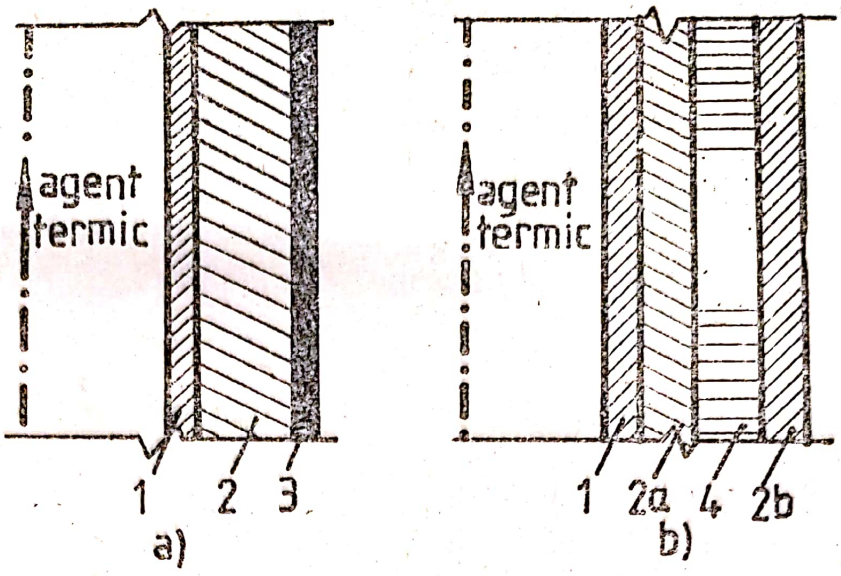


Fig. 6.4. Izolații termice analizate: *1*—conductă; *2a, 2b*—straturi termoizolante; *3*—strat protector; *4*—inele pîslă; *5* — strat de aer

Calculele au fost efectuate cu ajutorul programului ordinator ATEFS, pentru o conductă cu diametrul nominal $D=800$ mm, o temperatură a agentului termic de 150°C și temperatura mediului ambiant de 1°C.

În figura 6.5 se prezintă domeniul de analiză pentru varianta b) de izolație termică (secțiune longitudinală prin conductă), iar în figura 6.6 se arată distribuția de temperatură pentru condițiile date.

Se observă că izotermele nu sînt paralele cu suprafața conductei datorită neomogenității izolației termice, respectiv datorită distorsiunilor de cîmp termic generate de prezența spațiilor de aer. Se subliniază faptul că o astfel de distribuție nu poate fi obținută folosind calculul analitic, datorită neomogenității mediului studiat. Și chiar dacă mediul ar fi omogen, calculul bidimensional, folosind o analiză cu serii Fourier, ar fi fost greu de realizat.

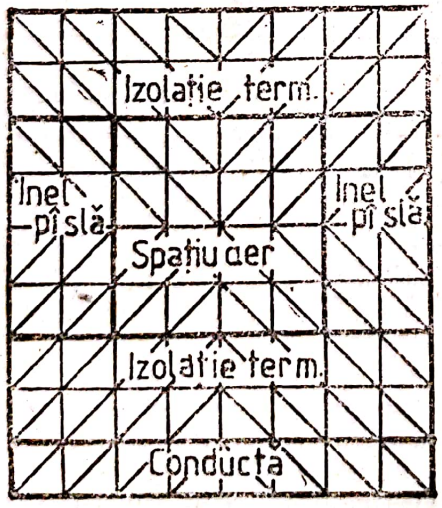


Fig. 6.5. Domeniul de analiză pentru varianta b)

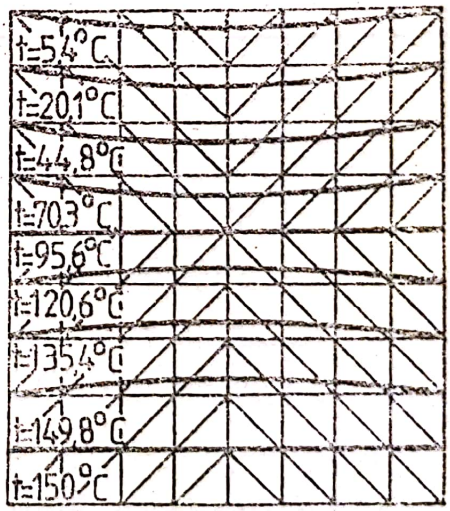


Fig. 6.6. Reprezentarea distribuției de temperatură pentru varianta b)

Scanned with OKEN Scanner

Aplicația 6.2.26

Se cere maximizarea, cu ajutorul algoritmului Simplex, a funcției obiectiv:

$$F=3x_1+5x_2+4x_3,$$

în prezența restricțiilor:

$$\begin{aligned} &2x_1+3x_2 \leqslant 8\\ &2x_2+5x_3 \leqslant 10\\ &3x_1+2x_2+4x_3 \leqslant 16\\ &x_1 \geqslant 0;\ x_2 \geqslant 0;\ x_3 \geqslant 0. \end{aligned}$$

În urma introducerii variabilelor auxiliare, sistemul anterior ia forma:

$$\begin{aligned} 2x_1+3x_2+\ \ \ x_4\ \ \ &=8\\ 2x_2+5x_3+\ x_5\ \ &=10\\ 3x_1+2x_2+4x_3+\ \ x_6&=16\\ x_1 \geqslant 0;\ x_2 \geqslant 0;\ x_3 \geqslant 0;\ x_4 \geqslant 0;\ x_5 \geqslant 0;\ x_6 \geqslant 0 \end{aligned}$$

Aplicarea algoritmului Simplex presupune următorii pași:

Pasul 1 (formarea tabelului inițial).

Funcția obiectiv și sistemul de ecuații transformat se scriu sub o formă tabelară (forma generală în tabelul 6.19 și particularizată pentru exemplul dat în tabelul 6.20), în care pentru prima bază s-au luat variabilele auxiliare introduse, iar funcția obiectiv s-a considerat scrisă sub forma:

$$F-3x_1-5x_2-4x_3=0$$

Tabelul 6.19

		$-x_1$	$-x_2$	...	$-x_n$	$-x_{n+1}$	$-x_{n+2}$	...	$-x_{n+m}$
x_{n+1}	b_1	a_{11}	a_{12}	...	a_{1n}	1	0	...	0
x_{n+2}	b_2	a_{21}	a_{22}	...	a_{2n}	0	1	...	0
:	:	:	:	...	:	:	:	...	:
x_{n+m}	b_m	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1
F	0	c_1	c_2	...	c_n	0	0	...	0

Tabelul 6.20

		$-x_1$	$-x_2$	$-x_3$	$-x_4$	x_5-	$-x_6$	Baza:
x_4	8	2	3	0	1	0	0	x_4, x_5, x_6
x_5	10	0	2	5	0	1	0	
x_6	16	3	2	4	0	0	1	
F	0	−3	−5	−4	0	0	0	

Se observă că funcția obiectiv nu conține variabilele bazei. Pentru simplificarea reprezentării tabelare, se suprimă vectorii unitari care aparțin variabilelor de bază, fapt ce nu schimbă cu nimic regulile de calcul. Ca urmare, tabelul inițial 6.19 poate fi reprezentat așa cum se indică în tabelul 6.21.

Scanned with OKEN Scanner

Pasul 2 (verificarea condiției de optimalitate).

Se verifică dacă soluția dată de tabel este optimă. În acest sens se inspectează coeficienții ultimului rînd al tabelului. Dacă toți acești coeficienți sînt pozitivi, s-a obținut soluția optimă. Dacă unul din ecoeficienți este negativ, algoritmul cere cel puțin o iterație în plus.

Tabelul 6.21

			$x_t \downarrow$			
			$-x_1$	$-x_2$	$-x_3$	Baza:
$x_s \leftarrow$	x_4	8	2	[3]	0	x_4, x_5, x_6
	x_5	10	0	2	5	
	x_6	16	3	2	4	
	F	0	−3	−5	−4	

Pasul 3 (determinarea variabilei x_t care intră în bază).

Se alege acea variabilă x_t din afara bazei (din cele scrise în partea superioară a tabelului), care îndeplinește condițiile:

— coeficientul c_j este cel mai mic negativ, adică:

$$c_t = \max | -c_j |;$$

— cel puțin unul din coeficienții de pe coloana variabilei x_t este pozitiv; dacă această condiție nu este îndeplinită, funcția obiectiv nu are optim, soluția fiind nemărginită.

Pasul 4 (determinarea variabilei x_s care iese din bază).

Alegerea variabilei x_s se face cu respectarea condiției:

$$\frac{b_s}{a_{st}} = \min_{a_{it}>0} \left\{\frac{b_i}{a_{it}}\right\}, \quad i=1, 2, \ldots, m; \; t>m,$$

adică pentru exemplul considerat:

$$\frac{b_s}{a_{st}} = \min\left\{\frac{8}{3}, \frac{10}{2}, \frac{16}{2}\right\} = \frac{8}{3} = \frac{b_1}{a_{12}}$$

Se marchează elementul pivot, obținut la intersecția coloanei corespunzătoare variabilei care intră în bază cu linia variabilei care iese din bază.

Pasul 5 (transformarea tabelului inițial).

Regulile de transformare ale tabelului sînt:

— se împart toți coeficienții liniei pivot cu elementul pivot a_{st};

— se scad apoi multiplii corespunzători noii linii pivot din celelalte linii ale tabelului, pentru a obține coeficienți zero în toate pozițiile coloanei pivot, folosind relația:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{sj}^{(k-1)} \frac{a_{it}^{(k-1)}}{a_{st}^{(k-1)}}$$

Se interschimbă în tabel pozițiile variabilelor x_t și x_s.

Se repetă pașii 2 și 5 pînă cînd în pasul 2 este îndeplinită condiția de optim.

Tabelul 6.22

		$\downarrow x_t$			
		$-x_1$	$-x_4$	$-x_3$	Baza:
x_2	8/3	2/3	1/3	0	x_2, x_5, x_6
x_5	14/3	−4/3	−2/3	[5]	
x_6	32/3	5/3	−2/3	4	
F	40/3	1/3	5/3	−4	

Scanned with OKEN Scanner

Tabelul 6.23

		$x_t \downarrow$		
		$-x_1$	$-x_4$	$-x_5$
x_2	8/3	2/3	1/3	0
$x_s \leftarrow x_3$	14/15	−4/15	−2/15	1/5
x_6	104/15	41/15	−2/15	−4/5
F	256/15	−11/15	17/15	4/5

Tabelul 6.24

		$-x_6$	$-x_4$	$-x_5$
x_2	40/41	−10/41	15/41	8/41
x_3	66/41	4/41	−6/41	5/41
x_1	104/41	15/41	−2/41	−12/41
F	776/41	11/41	45/41	24/41

Succesiunea operațiilor, în formă tabelară, pentru exemplul considerat, s-a prezentat în tabelele: 6.22, 6.23, 6.24, din care rezultă:

$$F_{max}=\frac{776}{41}=18{,}9268,$$

la valorile variabilelor:

$$x_1=\frac{104}{41}=2{,}5366;$$

$$x_2=\frac{40}{41}=0{,}9756;$$

$$x_3=\frac{66}{41}=1{,}6098;$$

$$x_4=x_5=x_6=0.$$

Rezolvarea acestei probleme se poate face folosind programul ordinator SIMPLEX.

Aplicația 6.2.27

Se aplică metoda programării dinamice la soluționarea problemei alegerii amplasamentului captării și a traseului unei magistrale de aducțiune a apei printr-o optimizare secvențială în caz determinist, discret.

Se consideră o magistrală de aducțiune pentru o localitate L, plecînd din două amplasamente de captare S_1 și S_2 (fig. 6.7). Traseele posibile trec prin punctele obligate A, B, C, D, E, formînd trei sectoare.

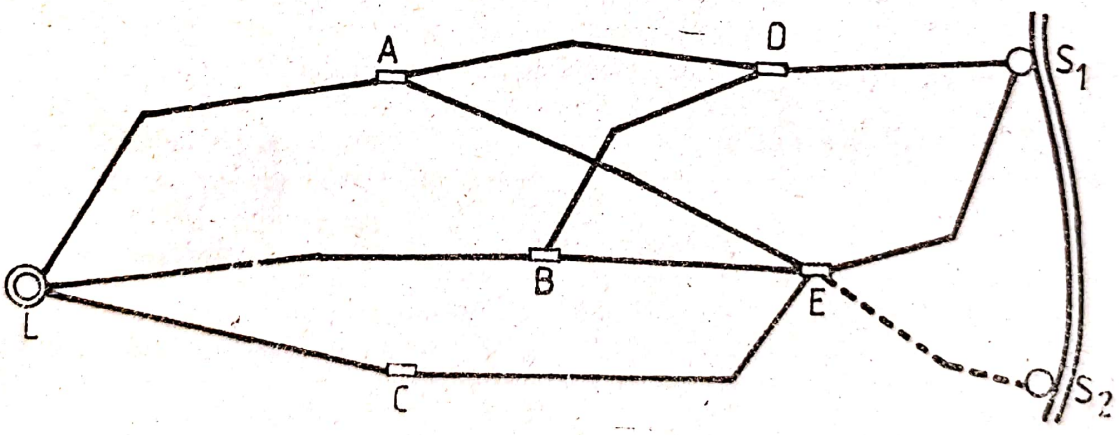


Fig. 6.7. Variante ale traseului aducțiunii

Punîndu-se problema stabilirii traseului pentru care costul total este minim, se determină pentru fiecare traseu investițiile parțiale și se întocmește graful secvențial din figura 6.8, unde fiecărui arc îi este asociat un cost, în unități convenționale. S-au notat cu x_0, x_1, x_2 și x_3 variabilele de decizie referitoare la fiecare sector. Aceste variabile nu vor lua valori numerice, ci vor reprezenta vîrfuri din graful respectiv ce se găsesc pe același aliniament.

Scanned with OKEN Scanner

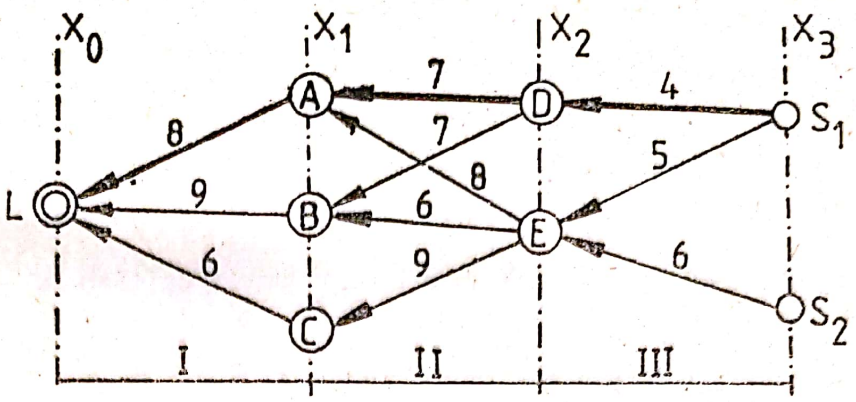


Fig. 6.8. Graful traseelor posibile ale aducțiunii

Se notează cu $V_I(x_0, x_1)$ costul sectorului I, ce depinde de valorile pe care le pot lua x_0 și x_1 (în cazul dat x_0 nu poate fi decît L). În mod identic se definesc $V_{II}(x_1, x_2)$ și $V_{III}(x_2, x_3)$.

Valoarea totală a sistemului de aducțiune se exprimă prin suma:

$$Z = V_I(x_0, x_1) + V_{II}(x_1, x_2) + V_{III}(x_2, x_3)$$

Inițial se consideră aducțiunea dezvoltată pe un singur sector (I) și notînd cu $f_I(x_1)$ costul minim al sectorului I, pentru fiecare din punctele obligate A, B, C, ecuația (4.62) devine:

$$Of_I(A) = V_I(L, A) = 8;\ f_I(B) = V_I(L, B) = 9;\ f_I(C) = V_I(L, C) = 6$$

peratorul „min" lipsește, deoarece nu există pasul zero.

Se consideră aducțiunea alcătuită din două sectoare (I, II), se notează cu $f_{I,\,II}(x_2)$ costul minim pentru sectoarele I și II împreună și pentru diferite valori ale lui x_2 ecuația (4.62) devine:

$$f_{I,\,II}(D) = \min_{x_1 = A, B, C} [V_{II}(x_1, D) + f_I(x_1)]$$

$$f_{I,\,II}(E) = \min_{x_1 = A, B, C} [V_{II}(x_1, E) + f_I(x_1)]$$

Dînd succesiv lui x_1 valorile A, B, C și ținînd seama că pentru legăturile inexistente se consideră valoarea ∞, din relațiile anterioare rezultă:

$$f_{I,\,II}(D) = \min[\underbrace{7+8}_{x_1=A},\ \underbrace{7+9}_{x_1=B},\ \underbrace{\infty+6}_{x_1=C}] = 15,\ \text{cu}\ x_1 = A;$$

$$f_{I,\,II}(E) = \min[\underbrace{8+8}_{x_1=A},\ \underbrace{6+9}_{x_1=B},\ \underbrace{9+6}_{x_1=C}] = 15,\ \text{cu}\ x_1 = C.$$

La ultimul pas se consideră că aducțiunea se dezvoltă pe sectoarele I, II și III și se scrie conform teoremei de optimalitate costul minim pentru sectoarele I, II și III împreună, corespunzător valorilor S_1 și S_2 ale lui x_3:

$$f_{I,\,II,\,III}(S_1) = \min_{x_2 = D, E} [V_{III}(x_2, S_1) + f_{I,\,II}(x_2)];$$

$$f_{I,\,II,\,III}(S_2) = \min_{x_2 = D, E} [V_{III}(x_2, S_2) + f_{I,\,II}(x_2)],$$

Scanned with OKEN Scanner

obținîndu-se pentru exemplul dat:

$$f_{I,II,III}(S_1)=\min[\underbrace{4+15}_{x_2=D},\ \underbrace{5+15}_{x_2=E}]=19,\ \text{cu}\ x_2=D;$$

$$f_{I,II,III}(S_2)=\min[\underbrace{\infty+15}_{x_2=D},\ \underbrace{6+15}_{x_2=E}]=21,\ \text{cu}\ x_2=E.$$

Rezultă deci, ca optimă soluția de amplasare a captării în S_1 cu traseul aducțiunii: S_1, D, A, L.

Întrucît procedeul de calcul analitic este mai puțin intuitiv, se prezintă un procedeu de calcul tabelar.

În tabelul 6.25 se înscrie în coloana 0 sectorul, în coloana 1 notația vîrfului care marchează începutul sectorului, în coloana 2 notația vîrfului care marchează sfîrșitul sectorului, în coloana 3 valoarea arcului corespunzătoare, în coloana 4 valoarea subpoliticii minime precedente, iar în coloana 5 suma coloanelor 3 și 4.

Pentru exemplificarea modului de întocmire a tabelului, se descrie cum se completează prima linie a sectorului II. În coloana 1, vîrful inițial este D, iar vîrful terminal este A, care se înscrie în coloana 2; în coloana 3 se completează valoarea acestui arc, adică 7; în coloana 4 valoarea subpoliticii minime precedente se citește din coloana 5, sectorul I, unde valoarea cea mai mică a politicii care începe cu A este 8; coloana 5 se completează ușor: $7+8=15$.

După completarea întregului tabel, se citește valoarea minimă din coloana 5 a sectorului III, care este 19, număr ce a fost încadrat într-un pătrat. De la aceasta, se pornește, cu o săgeată orizontal spre coloana 4 și se obține 15; de aici din nou în coloana 5 și în sus, la poziția de unde a fost transferat acest număr, ș.a.m.d. Săgețile arată soluția optimă. Notația vîrfurilor se citește în dreptul coloanei 1 pentru primul vîrf și în continuare în coloana 2 în dreptul săgeții orizontale: S_1, D, A, L (vîrfuri ce au fost subliniate în tabel).

Aplicația 6.2.28

Se exemplifică aplicarea algoritmului Bellman-Kalaba la determinarea traseului optim al magistralei de aducțiune pentru localitatea L, plecînd de la sursa S_1 (fig. 6.7).

Tabelul 6.25

Sector	x_n^i	x_{n+1}^i	$\lambda(x_n^i, x_{n+1}^i)$	Val subpol. min. preced.	(3+4)
0	1	2	3	4	5
I	A	$\underline{L}$	8	0 ←	8
	B	$\underline{L}$	9	0	9
	C	L	6	0	6
II	D	$\underline{A}$	7	8 ←	15
	D	$\underline{B}$	7	9	16
	E	A	8	8	16
	E	B	6	9	15
	E	C	9	6	15
III	S_1	$\underline{D}$	4	15 ←	19
	S_1	$\underline{E}$	5	15	20
	S_2	E	6	15	21

Scanned with OKEN Scanner

Modelarea problemei se realizează prin reprezentarea grafului conex orientat $G=(X, U)$ de ordinul $n=7$, alcătuit din sursă ca punct de origine, trasee ca arce și puncte obligate ca vîrfuri. Fiecărui arc $u_j^i \in U$ i se asociază un cost în unități convenționale (fig. 6.9), iar matricea M atașată grafului $G=(X, U)$ are elementele m_{ij} definite de relațiile (4.67):

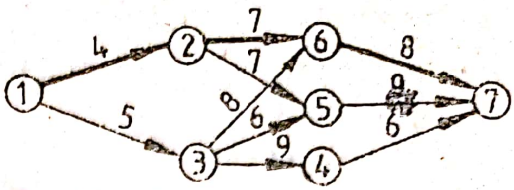


Fig. 6.9. Graful traseelor aducțiunii

	1	2	3	4	5	6	7	V_i^0	V_i^1	V_i^2	V_i^3
1	0	4	5	∞	∞	∞	∞	∞	∞	19	19
2	∞	0	∞	∞	7	7	∞	∞	15	15	15
3	∞	∞	0	9	6	8	∞	∞	15	15	15
$M=4$	∞	∞	∞	0	∞	∞	6	6	6	6	6
5	∞	∞	∞	∞	0	∞	9	9	9	9	9
6	∞	∞	∞	∞	∞	0	8	8	8	8	8
7	∞	∞	∞	∞	∞	∞	0	0	0	0	0

Traseul cu costul total minim este dat de drumul de valoare minimă în acest graf, ce se determină folosind algoritmul Bellman-Kalaba descris la pct. 4.3.3.2.

Pentru fiecare V_i^k $(k=0,1, \ldots)$ se adaugă la matricea precedentă cîte o coloană în care se trec valorile corespunzătoare.

Rezultă succesiv:

a) Se calculează valorile $V_i^0=m_{i6}$ $(i=1, \ldots, 7)$, care se trec în coloana V_i^0.

b) Se calculează valorile V_i^1 pentru $i=1, \ldots, 7$ $(j=1, \ldots, 7)$, ce se trec în coloana respectivă:

$$V_1^1=\min_{j\neq 1}(V_j^0+m_{1j})=\min(V_2^0+m_{12}, V_3^0+m_{13}, \ldots, V_7^0+m_{17})=\infty$$

$$V_2^1=\min_{j\neq 2}(V_j^0+m_{2j})=\min(V_1^0+m_{21}, V_3^0+m_{23}, \ldots, V_7^0+m_{27})=15$$

$$V_3^1=\min_{j\neq 3}(V_j^0+m_{3j})=\min(V_1^0+m_{31},\ V_2^0+m_{32}, \ldots, V_7^0+m_{37})=15$$

$$V_4^1=\min_{j\neq 4}(V_j^0+m_{4j})=\min(V_1^0+m_{41},\ V_2^0+m_{41}, \ldots, V_7^0+m_{47})=6$$

$$V_5^1=\min_{j\neq 5}(V_j^0+m_{5j})=\min(V_1^0+m_{51},\ V_2^0+m_{51}, \ldots, V_7^0+m_{57})=9$$

$$V_6^1=\min_{j\neq 6}(V_j^0+m_{6j})=\min(V_1^0+m_{61},\ V_2^0+m_{61}, \ldots, V_7^0+m_{67})=8$$

$$V_7^1=0$$

Scanned with OKEN Scanner

c) Se calculează valoriie V_i^2, pentru $i=1, \ldots,7$ $(j=1, \ldots, 7)$:

$$V_1^2=\min_{j\neq 1}(V_j^1+m_{1j})=\min(V_2^1+m_{12},\ V_3^1+m_{13},\ldots,V_7^1+m_{17})=19$$

$$V_2^2=\min_{j\neq 2}(V_j^1+m_{2j})=\min(V_1^1+m_{21},\ V_3^1+m_{23},\ldots,V_7^1+m_{27})=15$$

$$V_3^2=\min_{j\neq 3}(V_j^1+m_{3j})=\min(V_1^1+m_{31},\ V_2^1+m_{32},\ldots,V_7^1+m_{37})=15$$

$$V_4^2=\min_{j\neq 4}(V_j^1+m_{4j})=\min(V_1^1+m_{41},\ V_2^1+m_{42},\ldots,V_7^1+m_{47})=6$$

$$V_5^2=\min_{j\neq 5}(V_j^1+m_{5j})=\min(V_1^1+m_{51},\ V_2^1+m_{52},\ldots,V_7^1+m_{57})=9$$

$$V_6^2=\min_{j\neq 6}(V_j^1+m_{6j})=\min(V_1^1+m_{61},\ V_2^1+m_{62},\ldots,V_7^1+m_{67})=8$$

$$V_7^2=0$$

d) Pentru calculul valorilor corespunzătoare coloanei V_i^3; $i=1, \ldots, 7$ $(j=1,\ldots,7)$, se procedează analog:

$$V_1^3=\min_{j\neq 1}(V_j^2+m_{1j})=19$$

$$V_2^3=\min_{j\neq 2}(V_j^2+m_{2j})=15$$

$$V_3^3=\min_{j\neq 3}(V_j^3+m_{3j})=15$$

$$V_4^3=\min_{j\neq 4}(V_j^4+m_{4j})=6$$

$$V_5^3=\min_{j\neq 5}(V_j^5+m_{5j})=9$$

$$V_6^3=\min_{j\neq 6}(V_j^6+m_{6j})=8$$

$$V_7^3=0$$

Deoarece $V_i^2=V_i^3$ $(i=1, \ldots,7)$, algoritmul se oprește, iar valoarea drumului minim este dată de $V_1^2=V_1^3=19$. Această valoare este atinsă pe drumul (1, 2, 6, 7), rezultînd deci traseul optim al aducțiunii ca fiind: S_1, D, A, L.

Pentru soluționarea acestei probleme se poate utiliza programul ordinator DRUMIN.

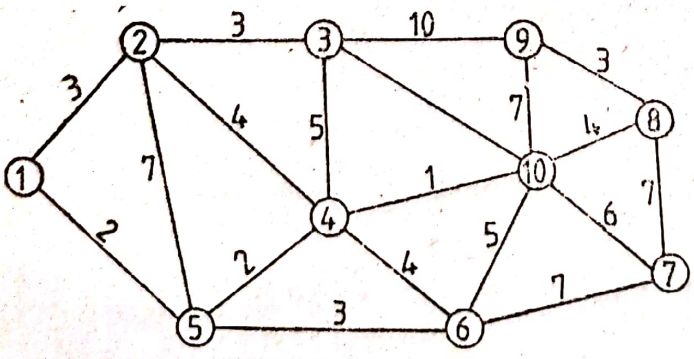


Fig. 6.10. Graf conex

Aplicația 6.2.29

Să se determine, folosind algoritmul Kruskal, arborele de valoare minimă al grafului conex din figura 6.10.

Se consideră inițial muchia [4, 10], cu cea mai mare valoare (egală cu 1) și se adaugă

Scanned with OKEN Scanner

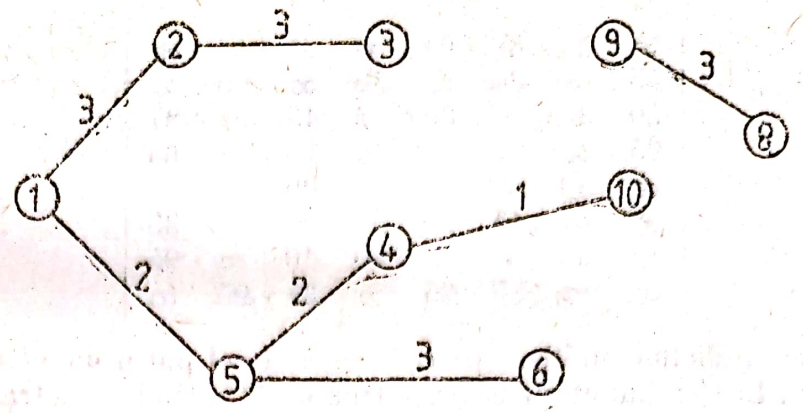


Fig. 6.11. Graf parțial

apoi succesiv, muchiile [1, 5], [5, 4] avînd valori egale cu 2 și neînchizînd cu [4, 10] nici un ciclu, precum și muchiile [1,2], [2,3], [5,6], [8,9], cu valori egale cu 3 și, de asemenea, neînchizînd nici un ciclu împreună cu muchiile alese anterior. Se obține astfel graful parțial din fig. 6.11, care nu este încă conex.

Se cercetează graful parțial din această figură și se încearcă să i se adauge o muchie cu valoarea egală cu 4. Muchia [2, 4] nu poate fi adăugată, deoarece ar închide ciclul [1, 2, 4, 5, 1] iar muchia [4, 6], de asemenea, deoarece ar închide ciclul [5, 4, 6]. În schimb se poate adăuga muchia [8, 10]. La graful parțial obținut astfel, se încearcă adăugarea unei muchii cu valoarea egală cu 5; muchiile [2, 4] și [6, 10] închid respectiv ciclurile [1, 2, 3, 4, 5, 1] și [5, 4, 10, 6], deci se încearcă cu muchii de valori mai mari (egale cu 6). Muchia [3, 10] închide ciclul [1, 2, 3, 10, 4, 5, 1], în schimb muchia [7, 10] nu închide nici un ciclu. Ca urmare, se adaugă muchia [7, 10], ceea ce conduce

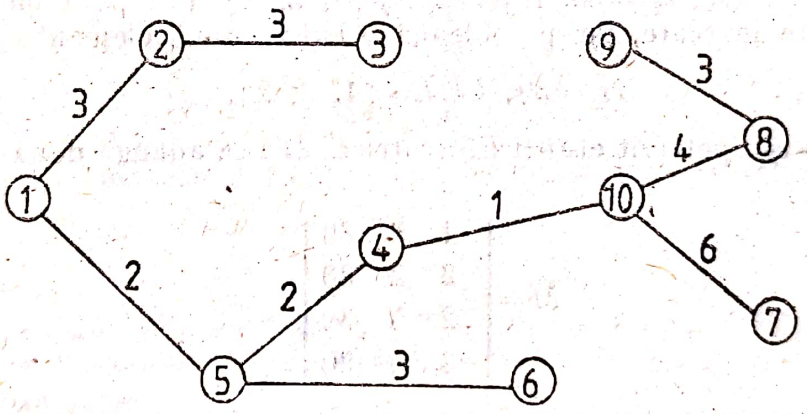


Fig. 6.12. Arbore minimal

a arborele minimal din fig. 6.12, avînd valoarea 27. La acest rezultat se poate ajunge cu ajutorul programului de calcul ARBMIN.

Aplicația 6.2.30

Se exemplifică aplicarea algoritmului descris la pct. 5.2.1 pentru determinarea traseelor optime ale unei rețele de conducte, avînd graful traseelor posibile de ordinul $n=8$ din figura 6.13, căruia i se atașează matricea costurilor:

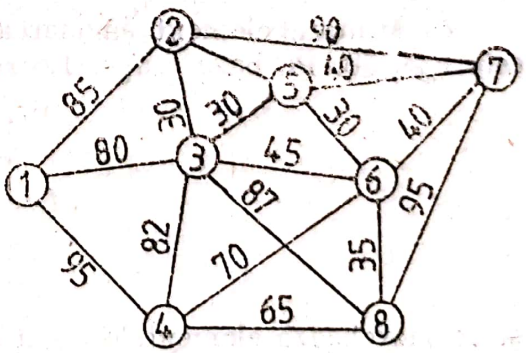


Fig. 6.13. Graful traseelor posibile ale rețelei de conducte

Scanned with OKEN Scanner

$$C=\begin{bmatrix} \infty & 85 & 80 & 95 & \infty & \infty & \infty & \infty \\ 85 & \infty & 30 & \infty & 30 & \infty & 90 & \infty \\ 80 & 30 & \infty & 82 & 30 & 45 & \infty & 87 \\ 95 & \infty & 82 & \infty & \infty & 70 & \infty & 65 \\ \infty & 30 & 30 & \infty & \infty & 30 & 40 & \infty \\ \infty & \infty & 45 & 70 & 30 & \infty & 40 & 35 \\ \infty & 90 & \infty & \infty & 40 & 40 & \infty & 95 \\ \infty & \infty & 87 & 65 & \infty & 35 & 95 & \infty \end{bmatrix}$$

Mulțimea indicilor liniilor pe care se află cel puțin un element marcat se notează cu L. Urmînd etapele alegoritmului de optimizare a traseelor rezultă succesiv:

a) Minimul elementelor de pe prima linie $c_{13}=80$ fiind unic, potrivit etapei E_2: $M=[1\ 3\ 80]$ și se marchează elementele c_{13}, c_{31}, în matricea C astfel că $L=\{1,3\}$.

b) Se determină minimul elementelor nemarcate de pe liniile 1 și 3 ale matricei C. Elementul minim nu este unic. Se alege c_{32}. Deoarece pe linia 2 a matricei C nu există elemente marcate potrivit etapei E5 pe linia următoare a lui M se înregistrează valorile 3, 2, 30, deci:

$$M=\begin{bmatrix} 1 & 3 & 80 \\ 3 & 2 & 30 \end{bmatrix}$$

și se marchează elementele c_{32}, c_{23} din matricea C, astfel că:

$$L=\{1\ \ 3,\ \ 2\}.$$

c) Acum minimul elementelor nemarcate de pe liniile ce conțin elemente marcate este $c_{25}=30$. Întrucît $5\notin\{1,\ 3,\ 2\}$ deci pe linia 5 a lui C nu sînt elemente marcate, iar pe coloana 5 există două elemente pentru care:

$$c_{k5}=30;\ \ k\in L=\{1,\ 3,\ 2\},$$

anume c_{25} și c_{35}, potrivit etapei E5 matricei M i se adaugă două noi linii, obținînd:

$$M=\begin{bmatrix} 1 & 3 & 80 \\ 3 & 2 & 30 \\ 2 & 5 & 30 \\ 3 & 5 & 30 \end{bmatrix}$$

Elementele ce se vor marca sînt c_{25}, c_{52}, c_{35} și c_{53} și ca urmare:

$$L=\{1,\ \ 3,\ \ 2,\ \ 5\}.$$

d) Minimul elementelor marcate de pe liniile de indici $r\in L=\{1,\ 3,\ 2,\ 5\}$ este $c_{56}=30$. Pe baza etapei E5 se poate scrie:

$$M=\begin{bmatrix} 1 & 3 & 80 \\ 3 & 2 & 30 \\ 2 & 5 & 30 \\ 3 & 5 & 30 \\ 5 & 6 & 30 \end{bmatrix}$$

și se marchează elementele c_{56} și c_{65} astfel că:

$$L=\{1,\ 3,\ 2,\ 5,\ 6\}.$$

Scanned with OKEN Scanner

e) În continuare se observă că minimul elementelor nemarcate de pe liniile de indici $r \in L=\{1, 3, 2, 5, 6\}$ este $c_{68}=35$. Deoarece pe linia 8 a lui C nu există elemente marcate, pe baza etapei E5 se completează matricea M cu o nouă linie:

$$M=\begin{vmatrix} 1 & 3 & 80 \\ 3 & 2 & 30 \\ 2 & 5 & 30 \\ 3 & 5 & 30 \\ 5 & 6 & 30 \\ 6 & 8 & 35 \end{vmatrix}$$

se marchează elementele c_{86} și c_{68} și se adaugă lui L valoarea 8:

$$L=\{1, \quad 3, \quad 2, \quad 5, \quad 6, \quad 8\}.$$

f) În continuare se găsește că $c_{57}=40$ este noul minim, iar cum pe linia 7 nu există elemente marcate și pe coloana 7 există două elemente pentru care:

$$c_{k7}=40; \qquad k \in L=\{1, \quad 3, \quad 2, \quad 5, \quad 6, \quad 8\},$$

anume c_{57} și c_{67}, efectuînd operațiile indicate de etapa E_5, matricea M devine:

$$M=\begin{vmatrix} 1 & 3 & 80 \\ 3 & 2 & 30 \\ 2 & 5 & 30 \\ 3 & 5 & 30 \\ 5 & 6 & 30 \\ 6 & 8 & 35 \\ 5 & 7 & 40 \\ 6 & 7 & 40 \end{vmatrix}$$

și se marchează elementele c_{57}, c_{75}, c_{67}, c_{76} ceea ce face ca mulțimea L să fie acum:

$$L=\{1, 3, 2, 5, 6, 8, 7\}.$$

g) Noua valoare minimă este $c_{36}=45$. Întrucît pe linia 6 există elemente marcate în virtutea etapei E5 se marchează elementele c_{36} și c_{63}, iar matricea M își păstrează forma.

h) Căutînd din nou valoarea minimă a elementelor nemarcate de pe liniile 1, 3, 2, 5, 6, 8 și 7 se găsește $c_{84}=65$. Deoarece pe linia 4 nu există încă un element marcat potrivit etapei E5, matricei M i se mai adaugă o linie obținînd:

$$M=\begin{vmatrix} 1 & 3 & 80 \\ 3 & 2 & 30 \\ 2 & 5 & 30 \\ 3 & 5 & 30 \\ 5 & 6 & 30 \\ 6 & 8 & 35 \\ 5 & 7 & 40 \\ 6 & 7 & 40 \\ 8 & 4 & 65 \end{vmatrix}$$

Scanned with OKEN Scanner

Se marchează elementele c_{84}, c_{48} și ca urmare:

$$L=\{1,\ 3,\ 2,\ 5,\ 6,\ 8,\ 7,\ 4\}.$$

i) Întrucît pe toate liniile matricei C există elemente marcate, calculele se opresc.

Așadar subgraful muchiilor optime este format din muchiile u_3^1, u_2^3, u_5^2, u_5^3, u_6^5, u_8^6, u_7^5, u_7^6, u_4^8. Costurile acestor muchii se află trecute în aceeași ordine, pe coloana a treia a matricei M. Deoarece pentru graful considerat $n=8$, iar matricea M are 9 linii, rezultă că problema arborelui minim are în acest exemplu mai multe soluții.

j) Se permută liniile matricei M astfel încît elementele coloanei a doua să fie ordonate crescător:

$$M=\begin{bmatrix} 3 & 2 & 30 \\ 1 & 3 & 80 \\ 8 & 4 & 65 \\ 2 & 5 & 30 \\ 3 & 5 & 30 \\ 5 & 6 & 30 \\ 5 & 7 & 40 \\ 6 & 7 & 40 \\ 6 & 8 & 35 \end{bmatrix}.$$

de pe coloana a doua a acestei matrice rezultînd frecvențele absolute ale celor $n-1=7$ vîrfuri ale grafului: $F_1=1$, $F_2=1$, $F_3=1$, $F_4=2$, $F_5=1$, $F_6=2$, $F_7=1$ și frecvențele cumulate: $F_0^*=0$, $F_1^*=1$, $F_2^*=2$, $F_3^*=3$, $F_4^*=5$, $F_5^*=6$, $F_6^*=8$, $F_7^*=9$.

Numărul arborilor minimi este dat de relația (5.33):

$$n_a=\prod_{i=1}^{7} F_i=1\cdot 1\cdot 1\cdot 2\cdot 1\cdot 2\cdot 1=4$$

iar elementele mulțimii $N_i=(F_{i-1}^*,\ F_i)\cap\{N\}$; $(i=1,\ \ldots,\ 7)$ sînt: $N_1=\{1\}$ $N_2=\{2\}$, $N_3=\{3\}$, $N_4=\{4,5\}$, $N_5=\{6\}$, $N_6=\{7,\ 8\}$, $N_7=\{9\}$ și ca urmare matricea ajutătoare are expresia:

$$A=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 5 & 5 \\ 6 & 6 & 6 & 6 \\ 7 & 8 & 7 & 8 \\ 9 & 9 & 9 & 9 \end{bmatrix}.$$

pe fiecare coloană a sa găsindu-se indicii liniilor matricei M ce conține caracteristicile muchiilor unuia din arborii optimi.

Astfel, elementele primei coloane a lui A precizează că unul din arborii minimi ai grafului este cel ce corespunde liniilor 1, 2, 3, 4, 6, 7, 9 din matricea M. Acest arbore este deci format din muchiile u_2^3, u_3^1, u_4^8, u_5^2, u_6^5, u_7^5, u_8^6 (fig. 6. 14,*a*). Analog se arată că ceilalți arbori sînt cei din figura 6.14,*b*, *c*, *d*. La acest rezultat se ajunge ușor folosind programul ordinator OTREDIRA.

Alegerea soluției optime ce se va aplica în practică se face ținînd seama și de alte criterii.

Scanned with OKEN Scanner

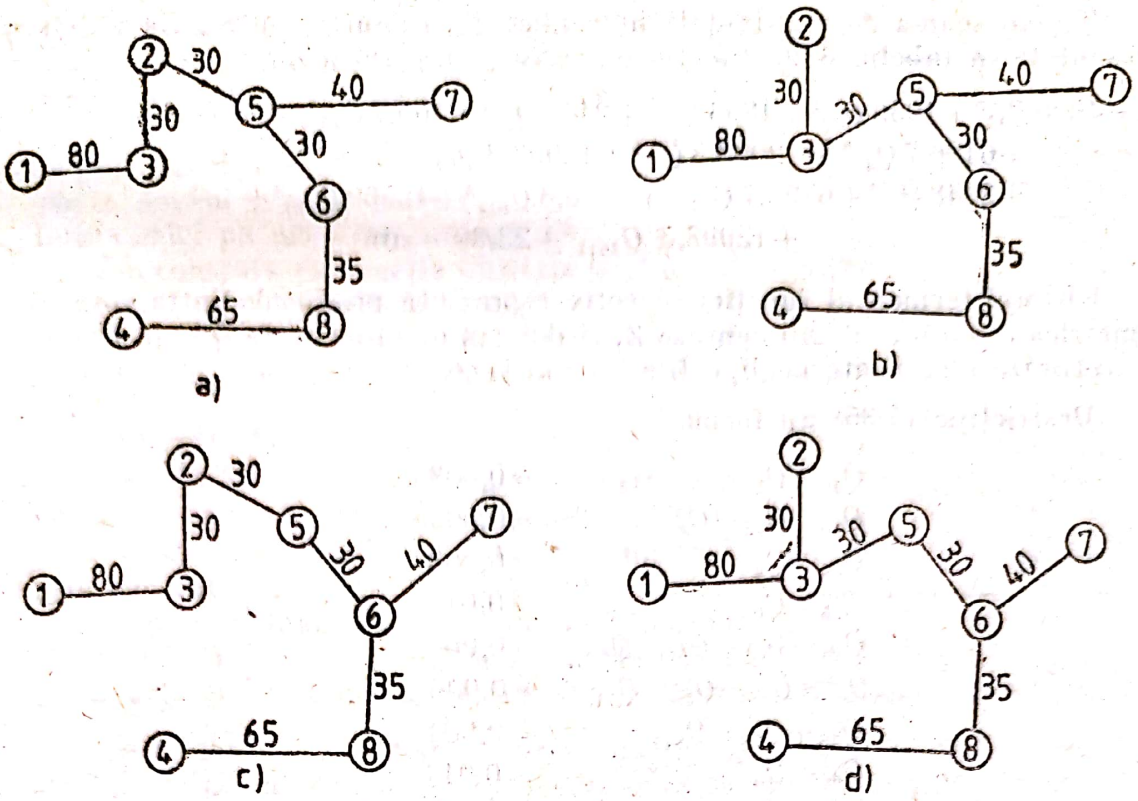


Fig. 6.14. Traseele optime ale rețelei de conducte

Aplicația 6.2.31

Se exemplifică aplicarea modelului „conținut" (MC), expus la pct. 5.2.2.3 pentru echilibrarea hidraulică a rețelei de distribuție a apei din figura 6.15, realizată din fontă, cunoscînd: lungimile L_{ij}, în m, diametrele D_{ij}, în mm, ale tronsoanelor, debitele industriale concentrate în nodurile 5, 9, 11, debitul de alimentare $Q=0{,}083$ m³/s, cota piezometrică în nodul de alimentare $Z_1=286{,}60$ m și exponentul $\beta=2$.

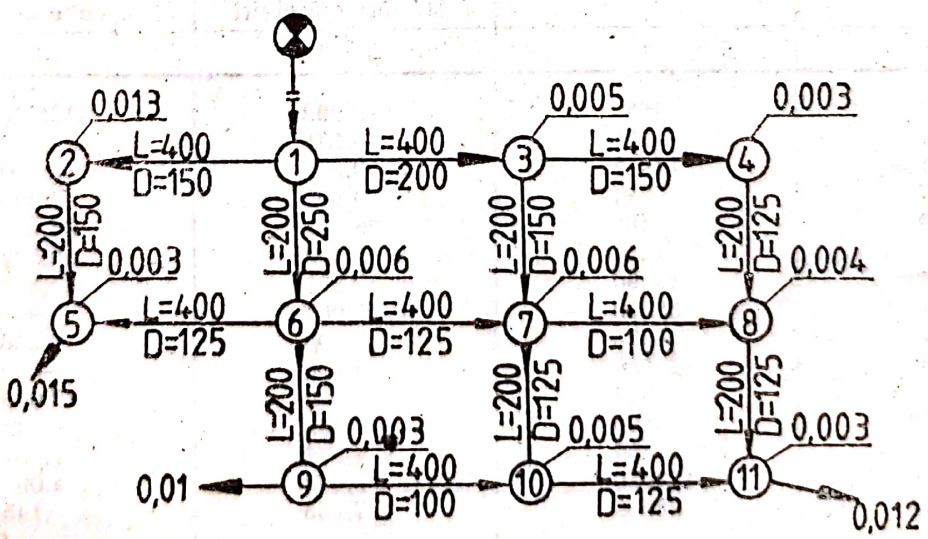


Fig. 6.15. Schema rețelei de alimentare cu apă

Scanned with OKEN Scanner

Ţinînd seama de rezistenţele hidraulice S_{ij} calculate cu relaţia (5.38) şi prezentate în tabelul 6.26, funcţia obiectiv (5.45) are forma:

$$F_e = 4896\,Q_{12}^3 + 1056\,Q_{13}^3 + 4896\,Q_{34}^3 + 2448\,Q_{25}^3 + 160{,}7\,Q_{16}^3 + 244\,Q_{37}^3 + \\ + 6496{,}7\,Q_{48}^3 + 12993{,}3\,Q_{65}^3 + 12993{,}3\,Q_{67}^3 + 42560\,Q_{78}^3 + \\ + 2448\,Q_{69}^3 + 6496{,}7\,Q_{710}^3 + 6496{,}6\,Q_{811}^3 + 42560\,Q_{910}^3 + \\ + 12993{,}3\,Q_{1011}^3 + 23{,}80 \rightarrow \min$$

Ultimul termen al funcţiei obiectiv reprezintă produsul dintre cota piezometrică a nodului de alimentare Z_1 şi debitul de alimentare Q şi în procesul de optimizare se poate neglija, fiind o constantă.

Restricţiile (5.35) au forma:

$$\begin{aligned}
Q_{12} - Q_{25} &= 0{,}003 \\
Q_{13} - Q_{34} - Q_{37} &= 0{,}005 \\
Q_{34} - Q_{48} &= 0{,}003 \\
Q_{25} - Q_{65} &= 0{,}018 \\
Q_{16} - Q_{35} - Q_{67} - Q_{69} &= 0{,}006 \\
Q_{37} - Q_{67} - Q_{78} - Q_{710} &= 0{,}006 \\
Q_{48} + Q_{78} - Q_{811} &= 0{,}004 \\
Q_{69} - Q_{9\ 10} &= 0{,}013 \\
Q_{78} + Q_{910} - Q_{1011} &= 0{,}005 \\
Q_{811} + Q_{1011} &= 0{,}015
\end{aligned}$$

Funcţia obiectiv împreună cu restricţiile conduc la o problemă neliniară, care s-a rezolvat cu ajutorul metodei gradienţilor condiţionaţi printr-un program ordinator, în limitele de precizie 10^{-5}.

În tabelul 6.26 sînt înscrise debitele pe tronsoane, iar în tabelul 6.27 se prezintă valorile cotelor piezometrice pe traseele 1—6—9—10—11 şi 1—3—4—8—11.

Tabelul 6.26

$i-j$	$S_{ij}[s^2/m^5]$	$Q_{ij}[dm^3/s]$	
		MC fără mărginiri	MC cu mărginiri
0	1	2	3
1—2	14688	12,992	12,996
1—3	3168	26,576	26,333
3—4	14688	10,873	10,810
2—5	7344	9,999	9,999
1—6	482	38,431	38,670
3—7	7344	10,703	10,522
4—8	19490	7,873	7,810
6—5	38980	8,007	8,000
6—7	38980	7,790	7,666
7—8	127680	4,060	4,043
6—9	7344	16,632	17,000
7—10	19490	8,432	8,145
8—11	19490	7,934	7,854
9—10	127680	3,632	4,000
10—11	38980	7,065	7,145
Funcţia obiectiv		$954{,}69 \cdot 10^{-4}$	$957{,}36 \cdot 10^{-4}$

Scanned with OKEN Scanner

Pentru a vedea evoluția funcției obiectiv s-au calculat și valorile debitelor pe tronsoane cînd s-a impus condiția $Q_{ij}>0{,}004$ m³/s.

De remarcat faptul că atunci cînd nu se impune un debit minim, rețeaua de distribuție poate deveni practic parțial ramificată (debite foarte mici pe unele tronsoane).

Se constată că funcția obiectiv are o valoare mai mare la aplicarea modelului „continut" cu mărginiri, iar neînchiderea liniei de presiune în nodul 11 este 0,036 m.

Tabelul 6.27

j	Z_j[m]
0	1
1	286,80
6	286,088
9	284,056
10	282,372
11	280,426

j	Z_j[m]
0	1
1	286,800
3	284,562
4	282,825
8	281,617
11	280,390

Aplicația 6.2.32

Se consideră rețeaua inelară exterioară de distribuție a apei cu topologia din figura 6.16 realizată din fontă și alimentată prin pompare cu debitul $Q_p=0{,}23$ m³/s, pentru care se cunosc: lungimile tronsoanelor L_{ij}, în m, cotele ZT_n la noduri și Z_{SP} la stația de pompare, în m, debitele concentrate q_j, în m³/s și presiunea necesară $HN_n=24{,}0$ m CA în nodurile de consum.

Se efectuează un studiu comparativ al dimensionării acestei rețele folosind modelul clasic al vitezelor medii economice (MVE) și modelele de optimizare descrise la pct. 5.2.3.4 și 5.2.3.5, respectiv modelul neliniar (MON) și modelul liniar (MOL), ultimul fiind aplicat atît în ipoteza consumului la noduri (MOL-N), cît și a consumului continuu uniform distribuit (MOL-D)

Calculul s-a făcut considerînd regimul de curgere a apei în conducte turbulent tranzitoriu, iar criteriul de optimizare adoptat a fost cel al consumului energetic minim.

Avînd în vedere că pentru debitul de alimentare a rețelei, diametrele tronsoanelor pot lua valori $D_{ij}=100...500$ mm, se consideră constantele $\lambda_*=0{,}01778$, $\beta=1{,}93$, ca valori medii ale valorilor extreme corespunzătoare limitelor posibile de variație ale diametrelor.

De asemenea, pentru parametrii economico-energetici s-au adoptat următoarele valori medii: $p_1=0{,}035$, $p_2=0{,}090$, $T_r=10$ ani, $\eta=0{,}70$, $f=5000$ lei/kW; $\sigma=1{,}3$, $\tau=0{,}95$, $\Phi_k=10{,}44$, $e=0{,}40$ lei/kWh.

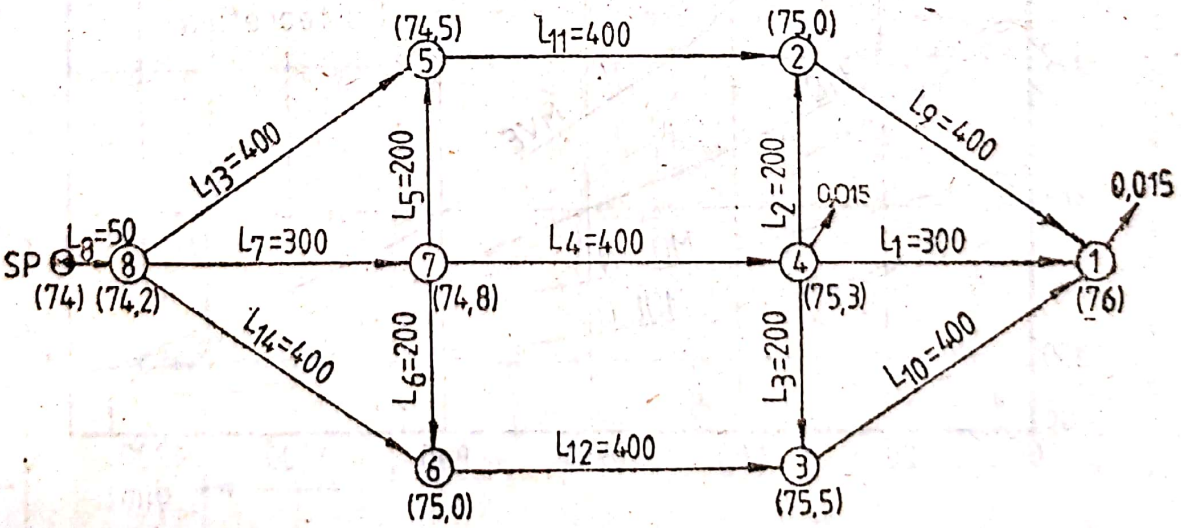


Fig. 6.16. Schema rețelei exterioare de distribuție proiectate

Scanned with OKEN Scanner

Tabelul 6.28

i–j	Modelul „MVE"				Modelul „MON"			
	Q_{ij} [m³/s]	D_{ij} [mm]	V_{ij} [m/s]	h_{ij} [m]	Q_{ij} [m³/s]	D_{ij} [mm]	V_{ij} [m/s]	h_{ij} [m]
0	1	2	3	4	5	6	7	8
4–1	0,006	100	0,76	1,43	0,0090	150	0,51	0.65
4–2	0,006	100	0,76	0,63	0,0026	125	0,30	0,10
4–3	0,006	100	0,76	0,63	0,0026	125	0,30	0,10
7–4	0,058	300	0,82	1,68	0,0541	300	0,75	0,81
7–5	0,006	100	0,76	0,93	0,0029	125	0,34	0,13
7–6	0,006	100	0,76	0,93	0,0029	125	0,34	0,13
8–7	0,100	350	1,04	1,20	0,0898	350	0,93	0,74
9–8	0,230	500	1,17	0,12	0,2300	500	1,17	0,12
2–1	0,017	200	0,54	0,80	0,0155	200	0,49	0,58
3–1	0,017	200	0,54	0,80	0,0155	200	0,49	0,58
5–2	0,031	250	0,63	1,37	0,0330	250	0,67	0,79
6–3	0,031	250	0,63	1,37	0,0330	250	0,67	0,79
8–5	0,050	250	1,02	2,13	0,0051	300	0,78	0,84
8–6	0,050	250	1,02	2,13	0,0051	300	0,78	0,84

Rezultatele calculului de dimensionare, efectuat cu calculatorul electronic folosind programele OPNELIN și OPDIREDA, referitoare la caracteristicile hidraulice ale conductelor, se prezintă în tabelele 6.28 și 6.29.

În figura 6.17 s-au reprezentat grafic, pornind de la stația de pompare *SP* la punctul obligat 1, pe traseul SP-8-5-2-1, liniile piezometrice obținute prin utilizarea celor trei modele de calcul menționate, punîndu-se în evidență

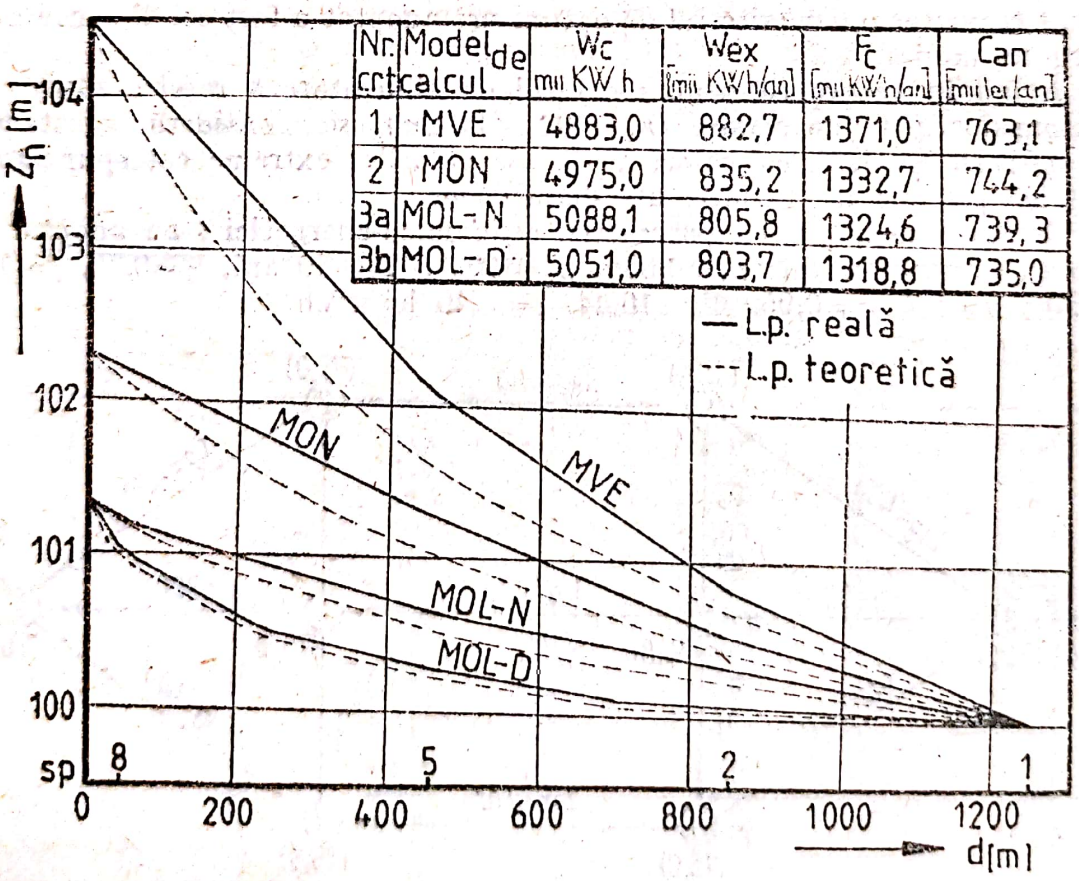


Fig. 6.17. Reprezentarea liniilor piezometrice pe traseul SP-8-5-2-1 și abaterea lor față de linia piezometrică optimă

Scanned with OKEN Scanner

Scanned with OKEN Scanner

Tabelul 6.29

$i-j$	Q_{ij}[m³/s]	Modelul „MOL — N"					Modelul „MOL — D"				
		Sector k	$X_{k,ij}$ [m]	$D_{k,ij}$ [mm]	$V_{k,ij}$ [m/s]	$h_{k,ij}$ [m]	Sector k	$X_{k,ij}$ [m]	$D_{k,ij}$ [mm]	$V_{k,ij}$ [m/s]	$h_{k,ij}$ [m]
0	1	2	3	4	5	6	7	8	9	10	11
4—1	0,0145	1	300	250	0,46	0,381	1 2	229 71	250 200	0,30 0,46	0,067 0,023
4—2	—0,0048	1	200	150	0,30	0,136	1	200	200	0,20	0,035
4—3	—0,0048	1	200	150	0,30	0,136	1	200	125	0,39	0,340
7—4	0,0457	1	400	300	0,65	0,584	1 2	228 172	350 250	0,48 0,93	0,120 0,510
7—5	0,0038	1	200	200	0,20	0,029	1	200	150	0,25	0,108
7—6	0,0038	1	200	200	0,20	0,029	1	200	200	0,20	0,019
8—7	0,0793	1 2	52 248	500 400	0,40 0,63	0,017 0,243	1 2	170 130	350 300	0,82 1,12	0,251 0,412
9—8	0,2300	1	50	500	1,17	0,123	1	50	400	1,83	0,245
2—1	0,0132	1	400	200	0,42	0,424	1	400	200	0,42	0,109
3—1	0,0132	1	400	200	0,42	0,424	1	400	250	0,30	0,127
5—2	0,0416	1 2	387 13	300 250	0,59 0,85	0,472 0,040	1 2	255 145	350 250	0,43 0,85	0,120 0,357
6—3	0,0416 0,0613	1 2	387 13	300 250	0,59 0,85	0,472 0,040	1	400	350	0,43	0,204
8—5	0,0613	1 2	315 85	400 350	0,49 0,64	0,189 0,099	1 2	225 175	400 300	0,49 0,87	0,107 0,412
8—6	0,0613	1 2	315 85	400 350	0,49 0,64	0,189 0,099	1	400	300	0,87	0,948

abaterea lor față de forma optimă teoretică și s-au înscris valorile corespunzătoare ale energiei înglobate în rețea W_c, ale energiei de exploatare W_{ex}, ale consumului energetic anual F_c, precum și ale cheltuielilor anuale C_{an}.

În urma studiului efectuat se constată că:

— față de dimensiunile obținute cu modelul clasic (MVE) se observă pe ansamblu o creștere a diametrelor conductelor la utilizarea modelelor de optimizare, întrucît modelul clasic nu ține seama de consumul minim de energie și de diversitatea parametrilor economici care determină costul de investiție și exploatare al rețelei;

— calculul diametrelor după modelele de optimizare a condus la rezultate mai economice, înregistrîndu-se o reducere a consumului energetic de realizare și exploatare a rețelei (MON→2,8%; MOL-N→3,3%; MOL—D→3,9%), precum și o însemnată economie de energie electrică de pompare (MON→7,1%; MOL—N→11,8%; MOL—D→12%);

— rezultatele optime obținute cu modelul liniar sînt superioare economico-energetic celor oferite de modelul neliniar;

— aplicarea modelului liniar de optimizare, în ipoteza consumului uniform distribuit, a condus la cele mai mici abateri față de forma optimă a liniei piezometrice, la cel mai mic consum energetic total și mai ales la o mai bună distribuție a energiei de pompare, prin eliminarea presiunilor disponibile sporite pe unele trasee chiar la consum maxim;

— reducerea în acest fel a presiunii în rețea are o deosebită importanță practică, contribuind implicit la micșorarea pierderilor de apă din sistem.

Pentru diferite rețele analizate, reducerea costului de investiție și exploatare, precum și economia de energie electrică pe seama micșorării pierderilor de sarcină prin aplicarea modelelor de optimizare, reprezintă valori de circa 3...25%.

Aplicația 6.2.33

Se ilustrează rezultatele aplicării modelului de optimizare expus la pct. 5.3 la dimensionarea rețelei interioare de alimentare cu apă rece a unei clădiri de locuit cu $P+2$ etaje (fig. 6.18), realizată din oțel, în care regimul de furnizare a apei este de 24 ore/zi.

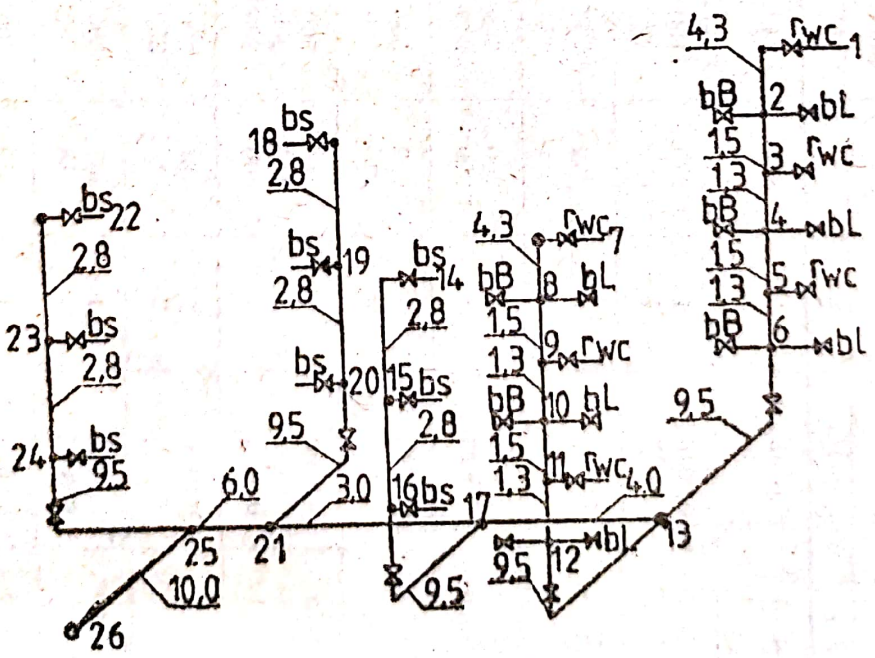


Fig. 6.18. Schema izometrică a rețelei interioare

Scanned with OKEN Scanner

Rezultatele calculului de dimensionare efectuat cu ajutorul programului ordinator DIREINTO, pentru următoarele valori medii ale parametrilor economico-energetici: $\beta_0=0,10$; $p_1=0,04$; $p_2=0,06$; $\eta=0,75$; $f=3000$ lei/kW; $\sigma=1,1$; $e=0,60$ lei/kWh; $\tau=0,35$, sînt prezentate în tabelul 6.30.

Acest model de dimensionare optimală prezintă avantajul că furnizează direct diametrele optime standardizate, permiţînd să se ţină seama la analiza soluţiilor optimale de reţele interioare de distribuţie a apei, de variaţia în timp a principalilor parametri economici, cum sînt: costul specific al energiei electrice, duratele de amortizare ale investiţiilor, costurile specifice ale materialelor etc., oferind posibilitatea unui calcul eficace în compararea variantelor constructive.

Aplicaţia 6.2.34

Se exemplifică aplicarea modelului de calcul descris la pct. 5.6 pentru dimensionarea optimală a izolaţiilor frigorifice la un perete exterior tencuit, orientat spre nord, cu dimensiunile $L=6$ m şi $H=4$ m, al unui spaţiu răcit la temperatura $t_i=-10°C$, funcţionînd tot timpul anului în zona de tempera-

Tabelul 6.30

Traseu	Tr	L_i [m]	ζ_i [–]	E_{ai} [–]	E_i [–]	q_i [l/s]	D_i mm	v_i [m/s]	h_i [m]	H_g [m]	H_u [m]	H_{nec} [mm]	D_0 [mm]
0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	4,3	22,5	0,50	0,05	0,075	15	0,4	0,32	1,3	2,0	2,00	—
	2	1,5	1,0	1,40	1,90	0,807	25	0,4	0,03	2,8	2,0	3,62	
	3	1,3	0,5	0,50	2,40	0,232	25	0,5	0,03	4,1	2,0	5,15	
	4	1,5	1,0	1,40	3,80	0,292	25	0,6	0,06	5,6	2,0	6,48	
	5	1,3	0,5	0,50	4,30	0,311	25	0,6	0,05	6,9	2,0	8,03	
	6	9,5	14,0	1,40	5,70	0,358	25	0,7	0,73	8,4	2,0	9,38	
	13	4,0	2,5	0,00	11,40	0,506	32	0,6	0,13	8,4	2,0	11,61	
	17	3,0	1,0	0,00	14,55	0,572	32	0,7	0,10	8,4	2,0	11,74	
	21	6,0	2,0	0,00	17,70	0,631	32	0,8	0,25	8,4	2,0	11,84	
	25	10,0	2,0	0,00	20,85	0,685	32	0,9	0,43	8,4	2,0	12,09	
2	7	4,3	22,5	0,50	0,50	0,075	15	0,4	0,32	1,3	2,0	2,00	—
	8	1,5	1,0	1,40	1,90	0,207	25	0,4	0,03	2,8	2,0	3,62	
	9	1,3	0,5	0,50	2,40	0,232	25	0,5	0,03	4,1	2,0	5,15	
	10	1,5	1,0	1,40	3,80	0,292	25	0,6	0,06	5,6	2,0	6,48	
	11	1,3	0,5	0,50	4,30	0,311	25	0,6	0,05	6,9	2,0	8,03	
	12	9,5	14,0	1,40	1,70	0,358	25	0,7	0,73	8,4	2,0	9,38	
3	14	2,8	17,0	1,05	1,05	0,154	15	0,9	0,94	2,8	2,0	2,00	17
	15	2,8	1,0	1,05	2,10	0,217	20	0,7	0,15	5,6	2,0	5,74	
	16	9,5	13,5	1,05	3,15	0,266	20	0,8	1,11	7,1	2,0	8,68	
4	18	2,8	17,0	1,05	1,05	0,157	15	0,9	0,94	2,8	2,0	2,00	16,2
	19	2,8	1,0	1,05	2,10	0,217	20	0,7	0,15	5,6	2,0	5,74	
	20	9,5	13,5	1,05	3,15	0,266	20	0,8	1,11	7,1	2,0	8,68	
5	22	2,8	20,5	1,05	1,05	0,154	15	0,9	1,07	2,8	2,0	2,00	15,4
	23	2,8	0,5	1,05	2,10	0,217	20	0,7	0,14	5,6	2,0	5,87	
	24	9,5	13,5	1,05	3,15	0,266	20	0,8	1,11	7,1	2,0	8,81	

Sarcina hidrodinamică necesară	$H_{nec}=12,52$ m
Costul de investiţie al conductelor	CC $=13,92$ mii lei
Costul anual al energiei de pompare	CE $=206,50$ mii lei/an
Cheltuieli anuale totale	CA $=2220,30$ mii lei/an

Scanned with OKEN Scanner

tură *IX* cunoscînd: $\delta_1=0{,}02$ m; $\lambda_1=0{,}93$ W/m °C; $\delta_2=0{,}03$ m; $\lambda_2=0{,}80$ W/m°C $w_f=0{,}0$ m/s; $w_e=4$ m/s; $p=0{,}05$; $T=10$ ani; $\beta_0=0{,}1$; $\sigma=1{,}1$; $\eta_e=0{,}3$; $\eta_f=0{,}6$ și la o conductă de refulare din oțel cu lungimea $L=15$ m ce transportă debitul de amoniac lichid $G=0{,}0224$ m³/s la temperatura $t_f=-10$°C cunoscînd: $T=10$ ani; $p=0{,}05$; $\sigma=1{,}1$; $\delta_p=20$ mm $\lambda_p=0{,}29$ W/m°C; $t_e=30$°C; $w=0{,}0$ m/s

S-a efectuat, cu ajutorul programelor ordinatoare DEFRIZOP și COF-; RIZOP, un studiu comparativ al dimensionării izolațiilor frigorifice aplicînd atît metoda clasică, cît și modelul de optimizare, luînd în considerare ca material izolant vata minerală (VM) și polistirenul expandat (PE), precum și mai mulți coeficienți ψ și admițînd mai multe valori c_f, pentru a evidenția tendința variației lor asupra grosimii izolației. Rezultatele sînt prezentate în tabelele 6.31 și 6.32.

Din analiza efectuată se constată următoarele:

— grosimea izolației determinată atît pe baza criteriului economic, cît mai ales pe cel energetic, este mai mare decît grosimea practicată curent;

— valorile foarte mari ale grosimii izolației obținute aplicînd exclusiv criteriul energetic de optimizare, se datorează ponderii mari a energiei necesare menținerii temperaturii în spațiul răcit față de energia înglobată;

— se înregistrează o variație lentă a grosimii optime a izolației cu costul energiei frigorifice;

Tabelul 6.31

c_f [lei/kWh]	δ_{iz} [mm]											
	Metoda clasică		Metoda optimizării									
			Crit. nerg.		Criteriul economico-energetic						Crit. econ	
			$\psi=0$		$\psi=0{,}25$		$\psi=0{,}50$		$\psi=0{,}75$		$\psi=1{,}00$	
	VM	PE	VM	PE	VM	PE	VM	PE	VM	PE	VM	PE
0	1	2	3	4	5	6	7	8	9	10	11	12
2,40	334	236	697	407	554	351	479	315	429	289	394	270
2,60					561	356	488	323	444	299	411	281
2,80					568	361	500	330	458	309	428	294
3,00					575	365	511	338	473	319	445	306

Tabelul 6.32

$\frac{d_e}{d_i}$ [mm/mm]	c_f [lei/kWh]	δ_{iz} [mm]											
		Metoda clasică		Metoda optimizării									
				Crit. energ.		Crit. economico-energetic						Crit. ec.	
				$\psi=0$		$\psi=0{,}25$		$\psi=0{,}50$		$\psi=0{,}75$		$\psi=1{,}00$	
		VM	PE	VM	PE	VM	PE	VM	PE	VM	PE	VM	PE
0	1	2	3	4	5	6	7	8	9	10	11	12	13
	0,60	120	100	435	360	335	265	315	250	295	230	220	165
219	0,65					335	265	310	250	300	230	225	170
203	0,70					340	270	320	290	300	235	235	175
	0,80					345	270	325	295	310	240	245	185

Scanned with OKEN Scanner

— prin aplicarea criteriului complex de optimizare, pentru o pondere mai mare a criteriului energetic și pentru ponderi egale ale ambelor criterii componente, valorile grosimii izolației sînt mari, situație ce se normalizează admițînd o pondere micșorată pentru criteriul energetic față de cel economic.

6.3. ELEMENTE DE PROGRAME ÎN LIMBAJUL FORTRAN

6.3.1. CONSIDERAȚII GENERALE

Limbajul de programare FORTRAN folosit în prezent în toate calculatoarele mari și mijlocii, precum și în multe microcalculatoare a apărut în anul 1956 și reprezintă o abreviere de la FORmula TRANslator. Este destinat în special pentru rezolvarea problemelor științifice sau inginerești.

Un program în FORTRAN trebuie scris pe linii ce cuprind maximum 80 coloane. Fiecare linie este destinată reprezentării unei informații pe primele 72 de poziții, ultimele 8 poziții fiind rezervate comentariilor. Dintre cele 72 de poziții active primele 5 sînt rezervate etichetelor instrucțiunilor care trebuie scrise aliniate spre dreapta. Poziția a 6-a este rezervată pentru a se indica, dacă este nevoie, că o instrucțiune se continuă și pe linia următoare.

Dacă se dorește să se folosească o linie din program pentru comentarii, avînd semnificații doar pentru programator, se poate realiza acest lucru scriind mai întîi în poziția 1 a acestei linii, litera *C*.

6.3.2. STRUCTURA PROGRAMELOR ÎN FORTRAN

Un program constă dintr-o succesiune de *instrucțiuni*. Se pot deosebi instrucțiuni executabile (care generează instrucțiuni în cod mașină) și instrucțiuni neexecutabile (care indică compilatorului natura și caracteristicile datelor de prelucrat, modul de prezentare a datelor pentru intrări-ieșiri etc). denumite de obicei, declarații.

Principalele tipuri de instrucțiuni sînt:

— instrucțiuni de atribuire, care au rolul de a indica modul în care se determină valorile variabilelor;

— instrucțiuni de salt, cu ajutorul cărora se stabilește o succesiune de executare a instrucțiunilor din program, alta decît cea secvențială;

— instrucțiuni de ciclare, care indică repetabilitatea unei anumite secvențe de program;

— instrucțiuni de intrare-ieșire, cu ajutorul cărora utilizatorul controlează transmiterea programului și a datelor către calculator și dinspre calculator;

— instrucțiuni pentru controlul lucrului cu subprograme.

Ordinea instrucțiunilor, pe tipuri, în cadrul unei unități de program, este:

— instrucțiunea de declarare a tipului unității de programe;

— declarații și alte instrucțiuni neexecutabile;

— instrucțiuni executabile (inclusiv STOP sau RETURN);

— instrucțiuni END.

Specificarea tipului unității de program este obligatorie doar pentru subprograme (tip funcție sau subrutină).

Instrucțiunea END semnalează compilatorului sfîrșitul unității de program.

Scanned with OKEN Scanner

6.3.3. ELEMENTELE DE BAZA ALE LIMBAJULUI FORTRAN

Alfabetul limbajului FORTRAN foloseşte 26 caractere literale: *A...Z* (alfabetul englez), 10 cifre arabe: 0...9 (sistemul zecimal), 4 simboluri aritmetice: +, −, *, /7 simboluri de punctuaţie: ., =, (,), $ şi spaţiile între caractere (blancuri).

Vocabularul limbajul FORTRAN este constituit dintr-un număr de aproximativ 80 de cuvinte cu semnificaţie bine stabilită (cuvinte cheie), precum şi dintr-un mare număr de cuvinte şi expresii construite de către programator. Cîteva cuvinte cheie mai frecvente sînt prezentate în tabelul 6.33.

Cuvintele cu semnificaţie variabilă formate de către utilizator sînt de două tipuri: identificatori şi etichete.

Un identificator este un grup de 1...6 caractere (litere sau cifre) dintre care primul este în mod obligatoriu o literă.

Identificatorii servesc la nominalizarea elementelor componente ale unei unităţi de program: variabile, subprograme etc. Identificatorii trebuie să aibă semnificaţie unică în cadrul unei unităţi de program. Într-un anumit program, identificatorii utilizaţi pentru nominalizarea subprogramelor nu pot fi folosiţi şi pentru variabile.

Etichetele sînt constituite din 1...5 cifre şi servesc la identificarea unei instrucţiuni.

Constantele limbajului FORTRAN sînt de trei tipuri: numerice, logice şi alfanumerice.

Constantele numerice pot fi: întregi, reale (zecimale) şi complexe.

Constantele întregi se scriu numai cu ajutorul cifrelor. Dacă în cazul numerelor pozitive prezenţa semnului (+) în faţa lor este facultativă, în cazul numerelor negative prezenţa semnului (−) în faţă este obligatorie. Constantele reale se scriu obligatoriu cu ajutorul punctului zecimal care are acelaşi rol ca virgula folosită în aritmetică. De exemplu, 0.31 reprezintă de fapt valoarea 0,31. În cea mai simplă formă de reprezentare a constantelor reale se pot folosi maximum 7 cifre semnificative.

Există doar două constante logice: .TRUE. şi .FALSE. (adevărat şi fals), ce pot fi asociate variabilelor sau expresiilor logice.

Tabelul 6.33

Cuvinte cheie

Cuvînt cheie	Semnificaţie	Cuvînt cheie	Semnificaţie
ABS	Valoare absolută	GO TO	Mergi la
BLOCK DATA	Unit. de programe conţinînd date	IABS	Valoare absolută întreagă
CALL	Apelează	IF	Dacă
COMMON	Comun	IMPLICIT	Implicit
CONTINUE	Continuă	INTEGER	Întreg
DATA	Date	LOGICAL	Logic
DIMENSION	Dimensiune	PAUSE	Întrerupe
DO	Execută	PRINT	Imprimă
DOUBLE		READ	Citeşte
PRECISION	Dublă precizie	REAL	Real (zecimal)
END	Sfîrşit	RETURN	Revino
EQUIVALENCE	Echivalenţă	STOP	Opreşte
FORMAT	Format	SUBRUTINE	Subrutină
FUNCTION	Funcţie	WRITE	Scrie

Scanned with OKEN Scanner

Constantele alfanumerice constau dintr-o succesiune de caractere alfanumerice cuprinse între apostrofuri sau precedate de descriptorul de format *wH*, în care: *w* este numărul caracterelor ce formează constanta (inclusiv blancurile), iar *H*-codul tipului de constantă.

Variabilele sînt mărimi ale căror valori se modifică în cursul executării programelor. Pentru reprezentarea variabilelor se folosesc identificatori.

Exemple de variabile scrise corect: *X*, *Y2*, NPLUSM și greșit: 2*M* (începe cu o cifră în loc de o literă), MATRICE (conține 7 litere, număr nepermis de mare).

Sînt permise trei categorii de variabile: numerice, logice și alfanumerice.

Variabilele numerice pot fi de tip întreg (INTEGER), real (REAL) sau complex (COMPLEX). Tipul unei variabile poate fi stabilit în trei moduri: prin convenție prestabilită, prin declarație de tip IMPLICIT sau utilizînd declarația de tip explicită. Declarațiile au prioritate față de convenție.

Conform convenției prestabilite, identificatorii care încep cu una din literele *I*, *J*, *K*, *L*, *M*, *N* sînt de tip întreg, în caz contrar sînt considerați de tip real.

Declarațiile de tip trebuie plasate înaintea primei instrucțiuni executabile a unității de program. Declarația de tip IMPLICIT precizează că tipul termenilor al căror identificator începe cu o literă conținută într-o listă din declarație este cel specificat înaintea listei respective:

IMPLICIT *tip 1* (*lista 1*), [*tip 2* (*lista 2*)]...

în care *tip 1*, *tip 2*,... sînt declaratori de tip (INTEGER, REAL, DOUBLE PRECISION, COMPLEX, LOGICAL), iar *lista 1*, *lista 2*,... sînt liste de litere (separate prin virgule).

Declarația de tip explicit indică tipul variabilelor după identificatori. Are forma generală:

Tip lista de identificatori

în care *tip* este un declarator de tip, iar *lista de identificatori* conține nume de variabile, tablouri sau funcții.

Pentru a putea interpreta variabilele algebrice cu indici, limbajul FORTRAN folosește variabile indexate. De exemplu, pentru a nu folosi 100 variabile distincte ca: *X*1, *X*2,..., *X*100, se folosește un nume comun pentru toate acestea, ele deosebindu-se prin indici diferiți. Deci, se poate defini variabila indexată *X* cu 100 componente și anume: *X*(1), *X*(2),..., *X*(100). Forma generală a instrucțiunii de declarare a variabilelor indexate este:

DIMENSION *idvi* (*listin*), [*idvi 1* (*listin 1*)]...

în care *idvi* este identificatorul variabilei, iar *listin* este lista variabilelor maxime ale indicilor.

Declararea variabilelor indexate se poate face și la declararea tipului.

Valoarea minimă pe care poate să o ia un indice este 1 (deci nu zero cum se obișnuiește în algebră). Pot fi utilizate și variabile indexate avînd doi sau mai mulți indici. De exemplu: *Y*(3,4) este componentă a variabilei indexate numită *Y* și anume componenta care în tabelul de variabile ocupă poziția situată la intersecția liniei a 3-a cu coloana a 4-a.

Indicii unei variabile indexate pot fi nu numai numere întregi, dar și variabile de tip întreg sau expresii formate cu acestea. De exemplu: *P*(*J*, *K*); *V1*(1.2**k*+1).

Scanned with OKEN Scanner

Expresiile sînt construcții de constante, variabile sau funcții, separate prin simboluri de operații aritmetice sau logice (eventual paranteze), care au un sens matematic sau logic.

Există trei tipuri de expresii: aritmetice, de relație, logice.

Expresiile aritmetice exprimă reguli de calcul conținînd următorii operatori: +(adunare); —(scădere); *(înmulțire); /(împărțire); **(ridicare la putere).

Pentru a codifica corect expresiile aritmetice, trebuie să se țină seama de regulile de prioritate ale operațiilor. Într-o expresie aritmetică care nu conține paranteze, operațiile sînt executate în ordinea următoare:

a) calcularea valorii funcției;
b) ridicarea la putere;
c) înmulțirea și împărțirea;
d) adunarea și scăderea.

Prin introducerea parantezelor (numai rotunde) se poate impune o altă ordine de executare a operațiilor.

Într-o expresie cu paranteze se vor executa mai întîi operațiile cuprinse între parantezele situate interior, iar în interiorul acestora se vor respecta prioritățile naturale. Operațiile cu aceeași prioritate se efectuează de la stînga la dreapta, cu excepția ridicării la putere care se execută de la dreapta la stînga.

O expresie de relație este constituită din două expresii aritmetice (tip I, *R* sau *D*) separate printr-un operator de relație. Simbolurile și semnificațiile operatorilor de relație sînt date în tabelul 6.34.

Tabelul 6.34

Operatori de relație

Operator	Exemple Scriere FORTRAN	Scriere algebrică
.LT.	A.LT.B	A < B
.LE.	A.LE.B	A ⩽ B
.EQ.	A.EQ.B	A = B
.NE.	A.NE.B	A ≠ B
.GT.	A.GT.B	A > B
.GE.	A.GE.B	A ⩾ B

O epresie logică este alcătuită dintr-una sau mai multe variabile sau expresii logice care sînt legate prin operatori logici, ca: .NOT. (nu este adevărat că); .AND. (și); .OR. (sau). În expresiile logice pot apare și paranteze.

Evaluarea unei expresii logice începe cu evaluarea expresiilor de relații, apoi executarea operațiilor logice cu ordinea de prioritate: .NOT., apoi .AND., apoi .OR. În caz de prioritate egală se execută operațiile de la stînga la dreapta.

6.3.4. PRINCIPALELE INSTRUCȚIUNI ALE LIMBAJULUI FORTRAN

a) Instrucțiuni de atribuire

Aceste instrucțiuni au ca efect calcularea valorii unei expresii și atribuirea acestei valori unei variabile. Există două tipuri de instrucțiuni de atribuire: aritmetice, logice.

Instrucțiunile de atribuire aritmetică sînt utilizate în vederea programării algoritmilor ce conțin calcule de tip algebric.

Scanned with OKEN Scanner

Se presupune de exemplu că într-un algoritm dat intervine secvența $x=\alpha+\beta$. Această secvență este preluată și programată în FORTRAN cu ajutorul unei instrucțiuni de atribuire astfel:

```
5|6|7
 | | X=ALFA+BETA
```

Se observă deci că s-au notat variabilele x, α, β, prin X, ALFA, respectiv BETA. Sensul acestei instrucțiuni este acela de evaluare a expresiei aritmetice din partea dreaptă a semnului egal, după care valoarea găsită este „atribuită" variabilei din partea stîngă a semnului egal, deci lui X. Această operație de „atribuire" explică de ce pot fi scrise instrucțiuni de tipul:

```
5|6| 7
 | | X=X+1
```

care în algebră nu au sens.

Într-o instrucțiune de atribuire aritmetică, în partea stîngă a semnului egal se găsește întotdeauna o variabilă, iar în partea dreaptă o expresie aritmetică.

O instrucțiune de atribuire logică este identică formal cu una aritmetică, cu deosebirea că expresia și variabila sînt de tip logic.

Pentru a atribui valori inițiale unor variabile se poate folosi instrucțiunea:

DATA *lisvar/liscon/*[, *lisvar 1/liscon 1* /]...

în care *lisvar*, *lisvar* 1 sînt liste de variabile, iar *liscon*, *liscon 1* sînt liste de constante. Dacă o constantă *con* trebuie să apară de *m* ori consecutiv, se poate scrie condensat *m*con*.

Elementele din blocuri comune (declarate prin COMMON) pot fi inițializate prin DATA doar în subprogramul BLOCK DATA.

b) Instrucțiuni de salt

Aceste instrucțiuni permit executarea instrucțiunilor unei unități de program în ordine diferită de ordinea în care acestea apar în memorie. Există patru categorii de instrucțiuni de salt:

— instrucțiuni de salt necondiționat;
— instrucțiuni de salt calculat;
— instrucțiuni de salt impus;
— instrucțiuni de salt condiționat.

Pentru a putea fi executată, instrucțiunea imediat următoare unei instrucțiuni de salt (excepție saltul calculat și saltul condiționat logic) trebuie etichetată.

Instrucțiunea de salt necondiționat. În general programul are o desfășurare secvențială, instrucțiune cu instrucțiune, pe măsura apariției acestora.

Introdusă în program atunci cînd îi vine rîndul să fie executată, instrucțiunea de salt necondiționat are ca efect transferul controlului desfășurării programului la o altă instrucțiune din program (deci nu la următoarea) avînd o etichetă identică cu cea care urmează după cuvîntul cheie GO TO.

Această instrucțiune are forma generală:

```
1...5|6|7
[n]| |GO TO n1
```

în care *n1* este eticheta instrucțiunii la care se face saltul.

Scanned with OKEN Scanner

Instrucţiunea de salt calculat are forma:

1...5	6	7	
[*n*]			GO TO (*n*1, *n*2, ..., *nm*) *i*

în care *n*1, *n*2,....*nm* sînt etichete de instrucţiuni executabile, iar *i* o variabilă întreagă, neindexată, care poate să ia valori între *1* şi *m*. Execuţia acestei instrucţiuni face ca următoarea instrucţiune care se execută să fie instrucţiunea cu eticheta *n*1 sau *n*2... sau *nm*, după cum variabila *i* are valoarea 1 sau 2 ... sau *m* în acel moment. Dacă *i* nu este între 1 şi *m* atunci se execută instrucţiunea următoare instrucţiunii GO TO.

Instrucţiunea de salt condiţionat IF aritmetic. Forma generală a acestei instrucţiuni este:

1...5	6	7	
[*n*]			IF (*e*) *n*1, *n*2, *n*3

în care: *e* este o expresie aritmetică evaluabilă, iar *n1*, *n2* şi *n3* sînt trei etichete de instrucţiuni. Dacă $e<0$ se face salt de la instrucţiunea cu eticheta *n*1; dacă $e=0$ se transferă execuţia la instrucţiunea cu eticheta *n*2, iar dacă $e>0$ se transferă execuţia la instrucţiunea cu eticheta *n*3.

Exemplu:

5	6	7
		.
		.
		.
15		$A=B+C$
		IF(X−Y) 15, 20, 25
20		$K=K+1$
25		EPS=SIN(*A*)

În momentul executării acestei instrucţiuni se evaluează mai întîi expresia aritmetică conţinută în paranteze, adică $X-Y$. În funcţie de valoarea găsită se efectuează apoi instrucţiunea cu eticheta 15, dacă valoarea este negativă, sau instrucţiunea cu eticheta 20, dacă expresia $X-Y$ este zero, sau instrucţiunea cu eticheta 25, dacă valoarea este pozitivă.

Instrucţiunea de salt condiţionat IF logic. Forma generală a acesteia este:

1...5	6	7	
[*n*]			IF (log) *iex*

în care *log* este expresia logică sau de relaţie, iar *iex* o instrucţiune executabilă, alta decît DO sau IF logic. Efectul acestei instrucţiuni este trecerea la execuţia instrucţiunii *iex* dacă expresia *log* are valoarea .TRUE., sau la execuţia instrucţiunii imediat următoare dacă expresia *log* are valoarea FALSE.

Exemplu:

5	6	7
		IF(I.LT. J) GO TO 5

c) Instrucţiunile STOP şi END

Sînt de obicei ultimele două instrucţiuni dintr-un program independent, în ordinea aceasta, semnalînd terminarea programului de calcul.

Scanned with OKEN Scanner

d) **Instrucțiunile DO și CONTINUE**

Aceste instrucțiuni în ordinea menționată semnalează începutul și sfîrșitul unui ciclu de calcule.

În mod obișnuit se formează un ciclu de calcul utilizînd instrucțiunile DO și CONTINUE astfel:

```
5 | 6 | 7
  |   | DO n K=n1, n2, n3
n |   | CONTINUE
```

în care: n este eticheta ultimei instrucțiuni din ciclu, în cazul de față CONTINUE; K-indicele contor al buclei; $n1$ — valoarea inițială a indicelui; $n2$ — valoarea finală a indicelui; $n3$ — rata de schimbare a indicelui (pas de variație).

Dacă valoarea $n3$ lipsește, se ia automat pasul de variație egal cu 1. Valorile $n1$, $n2$, $n3$ sînt constante sau variabile de tip întreg neindexate.

Exemplu:

```
5 | 6 | 7
  |   | DO 5 K=1,20
  |   | X(K)=0
  |   | Y(K)=0
5 |   | CONTINUE
```

Execuția acestui ciclu se face în felul următor: se atribuie mai întîi variabilei de control valoarea 1, apoi pentru această valoarea se execută toate instrucțiunile din interiorul ciclului, în cazul de față, două. Rezultă $X(1)=0$, $Y(1)=0$. Se transferă din nou controlul la începutul ciclului, mărindu-se valoarea variabilei contor cu o unitate. Se obține $K=2$. Se compară această valoare cu valoarea finală 20. Deoarece valoarea finală nu a fost atinsă se execută cele două instrucțiuni din interiorul ciclului, obținîndu-se $X(2)=0$, $Y(2)=0$. Se trece din nou controlul la începutul ciclului, deci $K=3$. Operațiile sînt reluate pînă cînd valoarea lui K mărită depășește valoarea finală 20. În acest moment controlul execuției programului trece la prima instrucțiune care succede linia finală a ciclului.

Instrucțiunea CONTINUE nu este obligatoriu a fi folosită pentru a marca sfîrșitul unui ciclu. Ea poate fi în general înlocuită de orice instrucțiune executabilă, cu excepția următoarelor instrucțiuni: DO, IF, GO TO, STOP, PAUSE, RETURN, CALL.

Se acceptă și două sau mai multe cicluri suprapuse avînd grijă ca indicii contor pentru aceste cicluri să fie notați diferit, dar folosind variabile de tip întreg. În cazul ciclurilor suprapuse execuția instrucțiunilor este următoarea: se începe cu execuția ciclului exterior; în momentul cînd se întîlnește un ciclu interior, acesta se execută complet și apoi se continuă execuția ciclului exterior; dacă ieșirea forțată din ciclu este posibilă (folosind o instrucțiune de salt condiționat sau necondiționat), intrarea forțată nu este permisă.

e) **Instrucțiunile WRITE și FORMAT**

Permit scrierea la imprimantă sau pe monitor a rezultatelor execuției programului de calcul. Forma generală a unei instrucțiuni de scriere cu format este următoarea:

```
5 | 6 | 7
  |   | WRITE (n, m) lisvar
m |   | FORMAT (lisdes)
```


Scanned with OKEN Scanner

în care: n este numărul simbolic al imprimantei sau monitorului; m — eticheta instrucțiunii FORMAT; *lisvar*—lista de variabile; *lisdes*—lista de descriptori, separarați de virgule, bare oblice (sleșuri), sau paranteze.

Descriptorii pentru date numerice au formele:

[a] Iw — pentru valori întregi;
]a] $Fw.d$ — pentru valori reale (forma fără exponent);
[a] $Ew.d$ — pentru valori reale (forma cu exponent);
[a] $Dw.d$ — pentru valori reale în dublă precizie;
]a] $Gw.d$ — pentru valori reale sau întregi;

unde: a este o constantă întreagă fără semn ce indică de cîte ori consecutiv se aplică descriptorul respectiv (în lipsă $a=1$); I, F, E, D, G — coduri de format; w — constantă întreagă fără semn care indică numărul de poziții ce sînt rezervate în vederea transmiterii informației; d — constantă întreagă fără semn, care indică numărul de poziții repartizate părții zecimale ($d<w$).

Formatele corespunzătoare instrucțiilor de scriere trebuie să conțină ca primă specificație, corespunzătoare scrierii unui rînd, un caracter de control care poate fi:

blanc — se trece la rîndul următor;
0 — se lasă un rînd liber și scrierea se face pe rîndul următor;
1 — se trece la pagina următoare;
\+ — tipărirea se face în același rînd;
/ — se trece la rîndul următor.

Exemplu:

```
      5 | 6 | 7
      ------------------------------------------
      |   |   | WRITE (3, 1) X
    1 |   |   | FORMAT (10X,' X=',F6.2)
```

Se observă că în cazul de față instrucțiunea FORMAT conține doar trei coduri separate prin virgule și anume:

— un cod de punere în pagină X, avînd prin forma sa, 10X, efectul ignorării primelor 10 poziții din rîndul ce urmează a fi scris;

— codul apostrof, utilizat pentru scrierea de mesaje și avînd în cazul de față efectul imprimării textului scris între cele două apostrofuri.

— descriptorul numeric F 6.2, care afectează o zonă de 6 poziții în care va fi imprimată valoarea variabilei X aliniată spre dreapta în cadrul zonei, ultimele două poziții fiind rezervate zecimalelor.

În final va apărea scris la perifericul de ieșire după 10 spații libere:

```
                                  x = — 8 . 7 2
| | | |
```

Exemplu:

```
     5 | 6 | 7
     ---------------------------------------
     |   |   READ (1, 1)(X(K), K=1,10)
   1 |   |   FORMAT (10F6.3).
```

Instrucțiunea FORMAT atașată lui READ indică modul cum trebuie făcută citirea și anume: se citesc cele 10 variabile $X(1)$, $X(2)$,..., $X(10)$, fiecare de pe un cîmp de 6 poziții, ultimele trei poziții fiind rezervate zecimalelor.

6.3.5. SUBPROGRAME

Limbajul FORTRAN admite trei tipuri de subprograme: de tip funcție externe și de tip BLOCK DATA.

6.3.5.1. Funcții

O funcție exprimă relația între variabile și eventual constante și/sau alte funcții.

Definirea unei funcții implică atribuirea unui nume, specificarea argumentelor acesteia și precizarea procedurii de calcul.

Se pot utiliza funcții predefinite (standard) și funcții definite de programator (funcții formulă).

Funcțiile standard (intrinseci) pot fi funcții interne (utilitare) și funcții externe (aritmetice sau de bibliotecă).

Pentru funcțiile elementare cele mai des întîlnite codificările sînt:

e^x — EXP(X)
ln x — ALOG(X)
log x — ALOG10(X)
sin x — SIN(X)
cos x — COS(X)
arctg x — ATAN(X)
$\sqrt{x}$ — SORT(X)

Toate aceste funcții acceptă un singur argument, de tip real și livrează un singur rezultat, de asemenea de tip real.

În ipoteza că se lucrează în dublă precizie, de regulă se adaugă la numele funcției litera *D* în față.

Funcțiile formulă se definesc în cadrul unității de program în care se utilizează astfel:

$$nff(p1[,p2]...)=e$$

în care: *nff* este numele funcției (identificator); *p1*, *p2*,... — parametrii formali; *e* — expresie de definire a funcției, ce poate să conțină ca operanzi:

- parametrii formali;
- constante (nu alfanumerice);
- variabile (nu alfanumerice);
- referiri la funcții intrinseci;
- referiri la funcții definite anterior;
- referiri la subprograme externe de tip FUNCTION.

Amplasarea subprogramelor funcție (intrinseci și definite) se face prin simpla utilizare ca operanzi în expresii.

Scanned with OKEN Scanner

Declarațiile de definire a funcțiilor se plasează înaintea instrucțiunilor executabile din unitatea de program respectivă și după declarațiile de tip pentru elementele ce o compun.

6.3.5.2. Subprograme externe

Un program extern constituie o unitate de program distinctă, fiind definit în mod extern față de unitatea de program care îl apelează. Este compus din instrucțiuni dintre care cel puțin una este executabilă. Prima instrucțiune trebuie să fie instrucțiune FUNCTION sau SUBROUTINE, iar ultima END.

Subprogramele de tip FUNCȚION se definesc astfel:

[*tip*] FUNCTION *nfe* (*p*1 [,*p*2]...)

în care: *tip* este tipul funcției (obligatoriu dacă nu este implicit); *nfe* — numele funcției; *p*1, *p*2,... — parametri formali, ce pot fi variabile neindexate, nume de variabile indexate, nume de procedură (cînd este obligatorie instrucțiunea EXTERNAL).

Numele funcției nu are voie să apară în altă instrucțiune neexecutabilă decît FUNCTION. Un astfel de subprogram nu poate conține instrucțiunile SUBROUTINE, BLOCK DATA sau o altă FUNCTION.

Apelarea se face utilizînd numele funcției (urmată de parametrii actuali) ca operand în cadrul unor expresii.

Parametrii formali nu pot apare în instrucțiunile EQUIVALENCE, COMMON sau DaATA.

Subprogramele de tip FUNCTION trebuie să conțină cel puțin o instrucțiune RETURN.

Subprogramele de tip SUBROUTINE se definesc astfel:

SUBROUTINE *nsr* [(*p*1 [, *p*2]...)]

în care *nsr* este numele subrutinei, iar *p*1, *p*2,... — parametri formali, de tipul celor dați la FUNCTION, plus *, ce permite revenirea în programul apelant în altă parte decît la instrucțiunea imediat următoare instrucțiunii CALL.

Apelul acestor subprograme se face prin instrucțiunea CALL. de forma:

CALL *nsr* [(*a*1 [*a*2]...)]

în care: *nsr* este numele subrutinei; *a1*, *a2*,... — parametri actuali cum ar fi constante, variabile, tablouri, sau de forma &*n*, unde *n* este eticheta unei instrucțiuni executabile, ce produce activarea unei subrutine.

Execuția unei instrucțiuni RETURN are ca efect revenirea în programul apelant la instrucțiunea imediat următoare instrucțiunii CALL, Execuția unei instrucțiuni RETURN *i* (*i* este constantă de tip întreg, pozitivă egală cel mult cu numărul asteriscurilor utilizate ca parametri în instrucțiunea SUBROUTINE) are ca efect revenirea în programul apelant la eticheta corespunzătoare celui de-al *i* — lea asterisc din instrucțiunea SUBROUTINE.

Pentru a se putea începe execuția subprogramelor FUNCTION și SUBROUTINE cu alte instrucțiuni decît plecînd de la linia de definiție, se utilizează instrucțiunea ENTRY, ce are forma:

ENTRY *npi* [(*p*1 [, *p*2]...)]

în care: *npi* este numele punctului de intrare în SUBROUTINE sau un nume de funcție în cazul subprogramelor FUNCTION; *p*1, *p*2, ...— parametri formali.

Scanned with OKEN Scanner

6.3.5.3. Subprograme BLOCK DATA

Subprogramele BLOCK DATA sînt utilizate pentru inițializarea blocurilor de date comune etichetelor. Instrucțiunea BLOCK DATA are forma:

BLOCK DATA *idbd*

în care: *idbd* este un identificator. În aceste subprograme sînt permise instrucțiunile: DIMENSION, DATA, COMMON, EQUIVALENCE, declarațiile de tip END (ultima instrucțiune).

Instrucțiunea COMMON permite accesul mai multor unități de programe la unele zone de memorie, denumite chiar diferit. Aceasta are structura:

COMMON[/[*nbloc 1*]/] lisvar 1 [[,]/[*nbloc* 2]/lisvar 2...]

în care: *nbloc* 1, *nbloc* 2,... sînt nume alese pentru blocurile considerate. Cînd *nbloc* lipsește se consideră că este vorba de blocul blanc (pentru care compilatorul folosește numele F%BLK). Dacă acesta este chiar primul bloc din declarație celor două bare se pot omite. Celelalte blocuri se numesc blocuri etichetate. Blocurile comune etichetate utilizate de mai multe unități de program trebuie să aibă aceeași lungime.

Ca elemente în listă pot fi utilizate variabile neindexate și/sau indexate.

Dacă identificatorul (numele) unui subprogram este utilizat ca parametru efectiv într-un program, acest identificator trebuie precizat într-o declarație EXTERNAL în programul apelat, înaintea tuturor instrucțiunilor executabile. Această instrucțiune are forma:

EXTERNAL *lisub*

în care *lisub* este o listă a identificatorilor de subprograme externe ce se vor utiliza ca parametri actuali.

6.3.6. LUCRUL CU FIȘIERE PE DISC FLEXIBIL ÎN CAZUL MICROCALCULATOARELOR

Se definește drept *fișier* un ansamblu de informații grupate și organizate într-un anumit mod, ce depinde de suportul fizic pe care se înmagazinează.

Fișierele pot conține date sau programe. Fișierele date pot conține la rîndul lor date numerice, alfanumerice sau texte.

În sistemele de operare CP/M și MS-DOS folosite pentru multe tipuri de microcalculatoare, numele unui fișier are următoarea structură:

[(dispozitiv):] (nume) [. (extensie)]

în care: *dispozitiv* reprezintă numele unității de disc în care se află fișierul specificat prin numele de fișier. Se folosesc notațiile: *A*: pentru unitatea de disc *A* și *B*: pentru unitatea de disc *B*, putînd exista pînă la 16 unități de discuri notate în continuare cu *C*:, *D*:, etc. Unitățile de hard-disc sînt desemnate de obicei începînd cu litera *C*; *nume* — numele propriu-zis al fișierului, ce poate avea pînă la 8 caractere, litere sau cifre, primul trebuind să fie literă; *extensie* — extensia numelui, compusă din pînă la 3 caractere (litere sau cifre), ce este opțională.

Dacă fișierul referit se află pe o unitate de disc instalată, dispozitivul poate lipsi din numele fișierului.

Scanned with OKEN Scanner

Exemplu:

A: OMEGA 12.A03, *B*: SIGMA. FOR, GAMA. REL, TEST. 1

Crearea fişierelor cu date de intrare se poate face cu programe create anume de utilizator în acest scop, sau cu programe ale sistemelor de operare folosite, denumite editoare de texte.

Fişierele de date create pe disc pot fi cu acces direct sau acces secvenţial, iar accesul la disc se poate realiza cu sau fără folosirea subrutinei OPEN a sistemelor de operare. Accesul direct permite scrierea într-un sau citirea dintr-un sector al fişierului.

a) **Operaţii de intrare-ieşire fără folosirea subrutinei OPEN**

Orice instrucţiune READ sau WRITE în care numărul unităţii logice al perifericului, NUL este 6...10, deschide automat un fişier pentru operaţii de citire respectiv de scriere de pe sau pe disc, fişierul rămînînd deschis pînă la întîlnirea unei instrucţiuni ENDFILE în sistemul CP/M ori CLOSE în sistemul MS-DOS, sau pînă la încheierea normală a programului.

Există următoarele restricţii:

— operaţia de intrare-ieşire se face pe discul din unitatea instalată;

— numele fişierului de date în sistemul CP/M este în mod obligatoriu FORT06.DAT... FORT10.DAT, corespunzător numărului unităţii logice, iar în sistemul MS-DOS este format din maximum 8 caractere, începînd cu o literă şi urmat de extensia .DAT. Instrucţiunile READ şi WRITE pot fi cu sau fără format.

O operaţiune de ieşire secvenţială la un fişier existent pe disc, îl şterge pe acesta şi deshide un alt fişier cu acelaşi nume. Deci, cu acces secvenţial nu este posibilă actualizarea fişierelor de pe disc.

Pentru accesul direct se foloseşte opţiunea REC=*n* în instrucţiunile READ şi WRITE. De exemplu instrucţiunile:

```
1...5 | 6 | 7
------------------------------------
      |   | K=30
      |   | WRITE (7, 20, REC=K) A, B, C
   20 |   | FORMAT (1X, 3F10.2)
```

scriu în fişierul FORT07. DAT de pe discul din unitatea instalată, în sectorul 30 al acestui fişier, numerele *A*, *B*, *C* conform instrucţiunii FORMAT asociate.

Instrucţiunile:

```
1...5 | 6 | 7
------------------------------------
      |   | DIMENSION A(20,15)
      |   |  .
      |   |  .
      |   | DO 18 I=16,20
   18 |   | READ (8, REC=I) (A(I, M), M=1,15)
```

determină citirea din fişierul FORT08.DAT de pe discul instalat a sectoarelor de la 16 la 20 şi încărcarea lor în matricea *A* în liniile 16...20. Citirea se face fără format, adică în binar.

b) **Operaţii de intrare-ieşire pe disc cu folosirea subrutinei OPEN**

Subrutina OPEN permite utilizatorului să folosească oricare din unităţile de disc, orice număr de unitate logică cuprins între 1 şi 10 şi orice nume de fişier. Accesul poate fi secvenţial sau direct, în acest din urmă caz folosind opţiunea REC.

Scanned with OKEN Scanner

Instrucțiunile READ și WRITE trebuie precedate de apelul subrutinei OPEN, care are forma:

— în sistemul CP/M:

CALL OPEN (*NUL*, '*nume de fișier*', *disc*)

— în sistemul MS-DOS:

OPEN (unit=*NUL*, file= '*nume de fișier. extenste*')

în care: *NUL* este numărul logic asociat fișierului cu valoarea între 1 și 10: *nume de fișier* — numele fișierului; *disc* — numărul unității de disc pe care se află fișierul, avînd valorile 0,1, 2, ..., pentru discul implicit, discul *A*, discul *B*,...

Dacă în CP/M numele de fișier are mai puțin de 11 caractere, numele se completează la sfîrșitul lui cu blancuri pînă la 11 caractere. Dacă numele de fișier are mai mult de 8 caractere, caracterele care depășesc acest număr constituie extensia (tipul) fișierului. Pentru diferite NUL, nume de fișiere și unități de disc, se dau următoarele exemple de apeluri ale subrutinei OPEN:

SISTEM	NUL	NUME DE FIȘIER	UNIT. DISC	INSTRUCȚIUNI DE APEL
CP/M	9	REZULTAT	A	CALL OPEN (9,'REZULTAT',1)
	3	TEMPERAT. URI	B	CALL OPEN (3,'TEMPERATURI',2)
MS-DOS	1	DIASTAN.DAT		OPEN (UNIT=1, FILE='DIASTAN.DAT')

Informații suplimentare privind sistemele de operare se găsesc în literatura de specialitate [63],]100].

Scanned with OKEN Scanner

```
C     ********************************
C     * P R O G R A M U L  A P R O L 1 *
C     ********************************
C     APROXIMAREA FUNCTIILOR MONOVARIABILE PRIN POLINOMUL LAGRANGE
C
C
C    LISTA NOTATIILOR
C      N-NUMARUL DE NODURI IN CARE SE CUNOSC VALORILE FUNCTIEI
C      M-NUMARUL DE NODURI IN CARE SE FACE INTERPOLAREA
C      X-VECTOR CONTININD PUNCTELE IN CARE SE CUNOSC VALORILE FUNCTIEI
C      F-VECTOR CONTININD VALORILE CUNOSCUTE ALE FUNCTIEI
C      Y-VECTOR CONTININD VALORILE IN CARE SE FACE INTERPOLAREA
C      H-VECTOR CONTININD VALORILE OBTINUTE,PENTRU FUNCTIE PRIN INTERPOLARE
C
C    ALOCAREA MEMORIEI
       DIMENSION X(100),F(100),Y(50),H(50)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
       READ (1,1) N,M
     1 FORMAT (2I5)
       READ (1,2) (X(K),K=1,N)
     2 FORMAT (7F10.4)
       READ (1,2) (F(K),K=1,N)
       READ (1,2) (Y(I),I=1,M)
       WRITE (3,3)
     3 FORMAT (////,2aX,/,32('*'),24X,'*P R O G R A M U L  A P R O L 1*',
      */,24X,32('*'))
       WRITE (3,4)
     4 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
       WRITE (3,5)
     5 FORMAT (/,29X,'VECTORII X(K) SI F(K)',/,29X,21('-'),//,25X,32('-'),
      */,25X,32('-'),/,26X,'K',10X,'X(K)',10X,'F(K)',/,25X,32('-'))
       DO 8 K=1,N
       WRITE (3,6) K,X(K),F(K)
     6 FORMAT (/,25X,I2,2(4X,F10.3))
       IF (K.NE.N) GOTO 8
       WRITE (3,7)
     7 FORMAT (25X,32('-'),//)
     8 CONTINUE
C
C    EFECTUAREA CALCULULUI SI SCRIEREA REZULTATELOR
       WRITE (3,9)
     9 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
       DO 12 I=1,M
       H(I)=0
       DO 12 K=1,N
       W=1
       DO 11 J=1,N
       IF (K-J) 10,11,10
    10 W=W*(Y(I)-X(J))/(X(K)-X(J))
    11 CONTINUE
    12 H(I)=H(I)+F(K)*W
       WRITE (3,13)
    13 FORMAT (/,19X,'VALORILE INTERPOLATE H(I) IN PUNCTELE Y(I)',
      */,19X,42('-'),//,25X,32('-'),/,26X,'I',10X,'Y(I)',10X,'H(I)'
      */,25X,32('-'))
       DO 15 I=1,M
       WRITE (3,6) I,Y(I),H(I)
       IF (I.NE.N) GOTO 15
       WRITE (3,14)
    14 FORMAT (25X,32('-'),//)
    15 CONTINUE
       STOP
       END
```

```
C     ********************************
C     * P R O G R A M U L  A P R O N *
C     ********************************
C     APROXIMAREA FUNCTIILOR PRIN METODA NEWTON
C
C
C    LISTA NOTATIILOR
C      N-NUMARUL DE NODURI IN CARE SE CUNOSC VALORILE FUNCTIEI
C      M-NUMARUL DE NODURI IN CARE SE FACE INTERPOLAREA
C      X-VECTOR CONTININD PUNCTELE IN CARE SE CUNOSC VALORILE FUNCTIEI
C      F-VECTOR CONTININD VALORILE CUNOSCUTE ALE FUNCTIEI
C      Y-VECTOR CONTININD VALORILE IN CARE SE FACE INTERPOLAREA
C      H-VECTOR CONTININD VALORILE OBTINUTE,PENTRU FUNCTIE  PRIN INTERPOLARE
C
C    ALOCAREA MEMORIEI
       DIMENSION X(100),F(100),Y(50),D(100,100),W(100),H(50)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
       READ (1,1) N,M
     1 FORMAT (2I5)
       READ (1,2) (X(K),K=1,N)
     2 FORMAT (7F10.4)
       READ (1,2) (F(K),K=1,N)
       READ (1,2) (Y(I),I=1,M)
       WRITE (3,3)
     3 FORMAT (////,25X,30('*'),/,25X,'*P R O G R A M U L  A P R O N*',
      */,25X,30('*'))
       WRITE (3,4)
     4 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
       WRITE (3,5)
     5 FORMAT (/,29X,'VECTORII X(K) SI F(K)',/,29X,21('-'),//,25X,32('-'),
      */,25X,32('-'),/,26X,'K',10X,'X(K)',10X,'F(K)',/,25X,32('-'))
       DO 8 K=1,N
       WRITE (3,6) K,X(K),F(K)
     6 FORMAT (/,25X,I2,2(4X,F10.3))
       IF (K.NE.N) GOTO 8
       WRITE (3,7)
     7 FORMAT (25X,32('-'),//)
     8 CONTINUE
C
C    EFECTUAREA CALCULULUI SI SCRIEREA REZULTATELOR
       WRITE (3,9)
     9 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
       DO 10 I=1,N
    10 D(I,1)=F(I)
       DO 12 K=2,N
       N1=N-K+1
       J=K
       DO 11 I=1,N1
       D(I,K)=(D(I,K-1)-D(I+1,K-1))/(X(I)-X(J))
    11 J=J+1
    12 CONTINUE
       DO 14 I=1,M
       W(1)=Y(I)-X(1)
       H(I)=W(1)*D(1,2)
       N2=N-1
       DO 13 K=2,N2
       W(K)=W(K-1)*(Y(I)-X(K))
       H(I)=H(I)+W(K)*D(K,K+1)
    13 CONTINUE
    14 H(I)=H(I)+F(1)
       WRITE (3,15)
    15 FORMAT (/,19X,'VALORILE INTERPOLATE H(I) IN PUNCTELE Y(I)',
      */,19X,42('-'),//,25X,32('-'),/,26X,'I',10X,'Y(I)',10X,'H(I)',
      */,25X,32('-'))
       DO 17 I=1,M
       WRITE (3,6) I,Y(I),H(I)
       IF (I.NE.N) GOTO 17
       WRITE (3,16)
    16 FORMAT (25X,32('-'),//)
    17 CONTINUE
       STOP
       END
```

```
C     ********************************
C     * P R O G R A M U L  A P R O G E *
C     ********************************
C     APROXIMAREA GEOMETRICA A UNEI FUNCTII PRIN METODA CELOR
C                    MAI MICI PATRATE
C
```


Scanned with OKEN Scanner

```
C
C    LISTA NOTATIILOR
C      N-NUMARUL PERECHILOR DE VALORI T-ZITA,CARE DESCRIU FUNCTIA CE
C        SE APROXIMEAZA PRIN CURBA DE FORMA ZITA=A0+A*T**B
C
C    ALOCAREA MEMORIEI
       DIMENSION T(150),ZITA(150)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
       READ (1,1) N,A0
     1 FORMAT (I5,F10.3)
       READ (1,2) ((T(J),ZITA(J)),J=1,N)
     2 FORMAT (5(F8.1,F8.2))
       WRITE (3,3)
     3 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  A P R O G E*',
      */,24X,32('*'))
       WRITE (3,4)
     4 FORMAT (///,22X,'VERIFICAREA PARAMETRILOR DE  INTRARE',/,22X,
      *35('='),//,25X,28('-'),/,27X,'I',2X,'J',1X,'I',4X,'T(J)',2X,'I',
      *2X,'ZITA(J)',1X,'I',/,25X,'I',26('-'),'I')
       DO 7 J=1,N
       WRITE (3,5) J,T(J),ZITA(J)
     5 FORMAT (25X,'I',2X,'I',/,25X,'I',I3,1X,'I',1X,F8.1,1X,'I',
      *1X,F8.2,1X,'I')
       IF (J.NE.N) GOTO 7
       WRITE (3,6)
     6 FORMAT (25X,28('-'))
     7 CONTINUE
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
       SX=0
       SY=0
       SXP=0
       SXY=0
       DO 8 J=1,N
       X=ALOG10(T(J))
       Y=ALOG10(ZITA(J)-A0)
       SX=SX+X
       SY=SY+Y
       SXP=SXP+X**2
       SXY=SXY+X*Y
     8 CONTINUE
       PSX=SX**2
       C=N*SXP-PSX
       C1=(SY*SXP-SX*SXY)/C
       C2=(N*SXY-SX*SY)/C
       A=10**C1
       B=C2
       WRITE (3,9)
     9 FORMAT (///,29X,'REZULTATELE CALCULULUI',/,29X,22('='))
       WRITE (3,10) A,B,A0
    10 FORMAT (//,29X,'COEFICIENTUL A=',F9.3,//,29X,'EXPONENTUL B=',F9.4,
      *//,29X,'COEFICIENTUL A0=',F9.3)
       STOP
       END



       ************************************
       * P R O G R A M U L  S I M P S O N *
       ************************************
       CALCULUL INTEGRALEI UNEI FUNCTII CU FORMULA LUI SIMPSON


       LISTA NOTATIILOR
         2N-NUMARUL DIVIZIUNILOR INTERVALULUI DE INTEGRARE
         A,B-LIMITELE INFERIOARA RESPECTIV SUPERIOARA DE INTEGRARE

       DEFINIREA FUNCTIEI
         F(X)=COS(1.570796*X*X)

       INTRODUCEREA DATELOR SI SCRIEREA LOR
         READ (1,1) N,A,B
       1 FORMAT (I5,2F8.3)
         WRITE (3,2)
       2 FORMAT (////,23X,34('*'),/,23X,'*P R O G R A M U L  S I M P S O N*',
        */,23X,34('*'))
         WRITE (3,3)
       3 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
         WRITE (3,4) N,A,B
       4 FORMAT (/,25X,'N=',I4,3X,'A=',F7.3,3X,'B=',F7.3)

       EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
         WRITE (3,5)
       5 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
         H=(B-A)/(2*N)
         S1=0
         S2=0
         DO 6 K=1,N
       6 S1=S1+F(A+2*K*H)
         N1=N+1
         DO 7 K=1,N1
       7 S2=S2+F(A+(2*K-1)*H)
         FS=(F(A)+4*S1+2*S2+F(B))*H/3
         WRITE (3,8) FS
       8 FORMAT (//,30 X,'INTEGRALA=',E12.6)
         STOP
         END

C      ********************************
C      * P R O G R A M U L  N E W C O T *
C      ********************************
C      CALCULUL INTEGRALEI UNEI FUNCTII CU FORMULA NEWTON-COTES
C
C
C      LISTA NOTATIILOR
C        N-NUMARUL NODURILOR ECHIDISTANTE
C        A,B-LIMITELE INFERIOARA RESPECTIV SUPERIOARA DE INTEGRARE
C        W-COEFICIENTII DE PONDERE COTES
C
C      DEFINIREA FUNCTIEI
         F(X)=1/(1+X)
C
C      ALOCAREA MEMORIEI
         DIMENSION W(25)
C
C      INTRODUCEREA DATELOR SI SCRIEREA LOR
         READ (1,1) N,A,B
       1 FORMAT (I5,2F8.3)
         READ (1,2) (W(I),I=1,N)
       2 FORMAT (6F10.6)
         WRITE (3,3)
       3 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  N E W C O T*',
        */,24X,32('*'))
         WRITE (3,4)
       4 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
         WRITE (3,5) N,A,B
       5 FORMAT (/,25X,'N=',I4,3X,'A=',F7.3,3X,'B=',F7.3)
         WRITE (3,6) (W(I) ,I=1,N)
       6 FORMAT (/,28X,'COEFICIENTII COTES W(I)',/,28X,23('-'),
        */,6(1X,F10.6))
C
C      EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
         WRITE (3,7)
       7 FORMAT (//,29X,'REZULTATELE CALCULUI',/,29X,22('*'))
         H=(B-A)/(N-1)
         FS=0
         N1=N-1
         DO 8 K=1,N1
       8 FS=FS+(B-A)*W(K)*F(A+K*H)
         WRITE (3,9) FS
       9 FORMAT (//,30X,'INTEGRALA=',E12.6)
         STOP
         END


C      ******************************
C      * P R O G R A M U L  E U L E R *
C      ******************************
C      REZOLVAREA ECUATIILOR DIFERENTIALE ORDINARE CU METODA EULER
C
C
C      LISTA NOTATIILOR
C        A,B-LIMITELE INTERVALULUI PE CARE SE DESFASOARA PROCESUL DESCRIS
C            DE ECUATIA DIFERENTIALA
```


Scanned with OKEN Scanner

```
C      H-PASUL CARE SEPARA DOUA VALORI CALCULATE PENTRU SOLUTIE
C      YO-VALOAREA INITIALA A SOLUTIEI CIND VARIABILA IA VALOAREA A
C      Y-VECTORUL CONTININD VALORILE SOLUTIEI
C
C     DEFINIREA FUNCTIEI
C       F(X,Y)=Y*SIN(X)+X
C
C     ALOCAREA MEMORIEI
        DIMENSION Y(1000)
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
        READ (1,1) A,B,H,YO
      1 FORMAT (4F10.3)
        WRITE (3,2)
      2 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  E U L E R*',
       */,24X,32('*'))
        WRITE (3,3)
      3 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
        WRITE (3,4) A,B,H,YO
      4 FORMAT (/,16X,'A=',F8.3,2X,'B=',F8.3,2X,'H=',F8.3,2X,'YO=',F8.3)
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
        WRITE (3,5)
      5 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'),
       */,27X,28('-'),/,32X,'X',15X,'Y',/,27X,28('-'))
        T=B-H
        K=1
        Y(K)=YO
      6 J=K+1
        X=A+(K-1)*H
        Y(J)=Y(K)+H*F(X,Y(K))
        WRITE (3,7) X,Y(J)
      7 FORMAT (/,23X,2(4X,F12.6))
        IF (X-T) 8,9,9
      8 K=K+1
        GOTO 6
      9 WRITE (3,10)
     10 FORMAT (27X,28('-'),//)
        STOP
        END
```

```
C      ********************************
C      *P R O G R A M U L  R U N K U T*
C      ********************************
C      REZOLVAREA ECUATIILOR DIFERENTIALE ORDINARE CU METODA
C                        RUNGE-KUTTA
C
C
C     LISTA NOTATIILOR
C       N-NUMARUL VALORILOR CALCULATE PENTRU SOLUTIE
C       A,B-LIMITA INFERIOARA RESPECTIV SUPERIOARA DE INTEGRARE
C       YO-VALOAREA INITIALA A SOLUTIEI IN PUNCTUL A
C       H-PASUL DINTRE DOUA VALORI CONSECUTIVE CALCULATE PENTRU
C         SOLUTIE
C       Y-VECTORUL CONTININD VALORILE SOLUTIEI
C
C     DEFINIREA FUNCTIEI
        F(X,Y)=Y+SIN(X)+X
C
C     ALOCAREA MEMORIEI
        DIMENSION X(1000),Y(1000)
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
        READ (1,1) N,A B,YO
      1 FORMAT (I5,3F10.3)
        WRITE (3,2)
      2 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  R U N K U T*'
       *,/,24X,32('*'))
        WRITE (3,3)
      3 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
        WRITE (3,4) N,A,B,YO
      4 FORMAT (/,19X,'N=',I2,2X,'A=',F8.3.2X,'B=',F8.3.2X,'YO=',F8.3)
C
C     EFECTUAREA CALCULULUI SI SCRIEREA REZULTATELOR
        WRITE (3,5)
      5 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
        H=(B-A)/(N-1)
        DO 6 K=1,N
        Y(K)=YO
      6 X(K)=A+(K-1)*H
        N1=N-1
        DO 7 K=1,N1
        C1=H*F(X(K),Y(K))
        C2=H*F(X(K)+H/2,Y(K)+C1/2)
        C3=H*F(X(K)+H/2,Y(K)+C2/2)
        C4=H*F(X(K)+H,Y(K)+C3)
        Y(K+1)=(C1+2*C2+2*C3+C4)/6+Y(K)
      7 CONTINUE
        WRITE (3,8)
      8 FORMAT (/,27X,28('-'),/,32X,'X',15X,'Y',/,27X,28('-'))
        DO 11 K=1,N
        WRITE (3,9) X(K),Y(K)
      9 FORMAT (/,23X,2(4X,F12.6))
        IF (K.NE.N) GOTO 11
        WRITE (3,10)
     10 FORMAT (27X,28('-'),//)
     11 CONTINUE
        STOP
        END
```

```
C      ********************************
C      *P R O G R A M U L  E C O J I N*
C      ********************************
C      REZOLVAREA APROXIMATIVA A ECUATIILOR FOLOSIND COMBINAT
C       METODA COARDEI SI METODA INJUMATATIRII INTERVALULUI
C
C
C     LISTA NOTATIILOR
C       A,B-LIMITELE INTERVALULUI DE DEFINIRE A FUNCTIEI F
C           PENTRU CARE SE CALCULEAZA APROXIMAREA RADACINII
C       EPS-EROAREA ADMISA IN CALCUL
C       K-IDENTIFICATOR CARE INDICA DACA S-A CALCULAT SAU NU
C         RADACINA ,DUPA CUM ARE VALOAREA 1 SAU 0
C       X-RADACINA ECUATIEI
C
C     ALOCAREA MEMORIEI
        EXTERNAL F
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
        READ (1,1) A,B,EPS
      1 FORMAT (2F10.4,F13.11)
        WRITE (3,2)
      2 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  E C O J I N*',
       */,24X,32('*'))
        WRITE (3,3)
      3 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
        WRITE (3,4) A,B,EPS
      4 FORMAT (/,20X,'A=',F10.4,2X,'B=',F10.4,2X,'EPS=',F13.11)
C
C     EFECTUAREA CALCULULUI SI SCRIEREA REZULTATELOR
        WRITE (3,5)
      5 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
        CALL RADEC (F,A,B,EPS,K,X)
        IF (K.EQ.0) GOTO 7
        WRITE (3,6) X
      6 FORMAT (//,24X,'RADACINA ECUATIEI: X=',F10.5)
        GOTO 9
      7 WRITE (3,8)
      8 FORMAT (//,23X,'FUNCTIA ARE ACELASI SEMN LA CAPETE')
      9 STOP
        END

        SUBROUTINE RADEC (F,A,B,EPS,K,X)
        K=1
        A1=A
        B1=B
        IF (A2*B2) 4,2,1
      1 K=0
        RETURN
```


Scanned with OKEN Scanner

```
      2 IF (A2,EQ,0)
        GOTO 3
        X=B1
        RETURN
      3 X=A1
        RETURN
      4 X1=0,5*(A1+B1)
        X11=F(X1)
        X2=A1-(B1-A1)*A2/(B2-A2)
        X21=F(X2)
        IF (X2,GE,X1) GOTO 5
        X=X1
        X1=X2
        X2=X
        X=X11
        X11=X21
        X21=X
      5 IF (A2*X11) 6,7,8
      6 IF (X1-A1,LT,EPS) GOTO 7
        B1=X1
        B2=X11
        GOTO 4
      7 X=X1
        RETURN
      8 IF (B2*X21) 9,10,11
      9 IF (B1-X2,LT,EPS) GOTO 10
        A1=X2
        A2=X21
        GOTO 4
     10 X=X2
        RETURN
     11 A1=X1
        A2=X11
        B1=X2
        B2=X21
        GOTO 4
        END

        FUNCTION F(X)
        F=X**3-0.5*X+4
        RETURN
        END

C       ******************************
C       * P R O G R A M U L  E C N E W *
C       ******************************
C       REZOLVAREA AFROXIMATIVA A ECUATIILOR CU METODA NEWTON
C
C
C     LISTA NOTATIILOR
C       X0-AFROXIMATIA INITIALA A RADACINII ECUATIEI
C       EPS-EROAREA ADMISA IN CALCUL
C       NI-NUMARUL MAXIM DE ITERATII
C       X-RADACINA ECUATIEI
C
C     DEFINIREA FUNCTIEI SI A DERIVATEI
        F(X)=EXP(X)+SIN(X)+X
        FP(X)=EXP(X)+COS(X)+1
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
        READ (1,1) NI,X0,EPS
      1 FORMAT (I5,F10.6,F10.7)
        WRITE (3,2)
      2 FORMAT (////,25X,30('*'),/,25X,'*P R O G R A M U L  E C N E W*',
       */,25X,30('*'))
        WRITE (3,3)
      3 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
        WRITE (3,4) NI,X0,EPS
      4 FORMAT (/,28X,'NI=',I3,2X,'X0=',F10.6,2X,'EPS=',F10.7)
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
        WRITE (3,5)
      5 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
        ITER=1
      6 X=X0-F(X)/FP(X)
        IF(ABS(X-X0).LE.EPS) GOTO 8
        ITER=ITER+1
        IF(ITER.LE.NI) GOTO 7
        GOTO 10
      7 X0=X
        GOTO 6
      8 WRITE (3,9) X
      9 FORMAT (//,24X,'RADACINA ECUATIEI: X=',F10.6)
        GOTO 12
     10 WRITE (3,11) NI
     11 FORMAT (//,22X,'PROCESUL NU CONVERGE IN ',I3,' ITERATII')
     12 STOP
        END



C       ********************************
C       * P R O G R A M U L  I N V M A P *
C       ********************************
C       INVERSAREA MATRICELOR PATRATICE PRIN METODA PARTITIONARII
C
C
C     LISTA NOTATIILOR
C       N-DIMENSIUNEA MATRICEI A CARE SE INVERSEAZA
C       D-INVERSA MATRICEI A
C
C     ALOCAREA MEMORIEI
        DIMENSION A(100,100),D(100,100),B(100,100),X(100),Y(100)
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
        READ (1,1) N
      1 FORMAT (I5)
        DO 2 I=1,N
      2 READ (1,3) (A(I,J),J=1,N)
      3 FORMAT (10F7.3)
        WRITE (3,4)
      4 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  I N V M A P*',
       */,24X,32('*'))
        WRITE (3,5)
      5 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
        WRITE (3,6)
      6 FORMAT (/,35X,'MATRICEA A',/,35X,10('-'))
        DO 7 I=1,N
      7 WRITE (3,8) (A(I,J),J=1,N)
      8 FORMAT (/,13X,8F7.3)
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
        WRITE (3,9)
      9 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
        DO 10 I=1,N
        DO 10 J=1,N
     10 D(I,J)=0
        D(1,1)=1/A(1,1)
        DO 13 K=2,N
        CALL BETA (N,A,D,K,BT)
        CALL CX (N,A,D,K,BT,X)
        CALL VY (N,A,D,K,BT,Y)
        CALL TB (N,X,Y,K,BT,B)
        K1=K-1
        DO 12 I=1,K1
        DO 11 J=1,K1
     11 D(I,J)=D(I,J)+B(I,J)
        D(I,K)=X(I)
     12 D(K,I)=Y(I)
     13 D(K,K)=BT
        WRITE (3,14)
     14 FORMAT (/,29X,'INVERSA D A MATRICEI A',/,29X,22('-'))
        DO 15 I=1,N
     15 WRITE (3,16) (D(I,J),J=1,N)
     16 FORMAT (/,5X,8(F7.3,1X))
        STOP
        END

        SUBROUTINE BETA (N,A,D,K,BT)
        DIMENSION A(100,100),D(100,100),T(100)
        K1=K-1
        H=0
        DO 2 I=1,K1
```


Scanned with OKEN Scanner

```
      S=0
      DO 1,I=1,K1
    1 S=S+A(K,J)*D(J,I)
      T(I)=S
    2 H=H+T(I)*A(I,K)
      BT=1/(A(K,K)-H)
      RETURN
      END

      SUBROUTINE CX (N,A,D,K,BT,X)
      DIMENSION A(100,100),D(100,100),X(100)
      K1=K-1
      DO 2 I=1,K1
      S=0
      DO 1 J=1,K1
    1 S=S+D(I,J)*A(J,K)
    2 X(I)=-BT*S
      RETURN
      END
      SUBROUTINE VY (N,A,D,K,BT,Y)
      DIMENSION A(100,100),D(100,100),Y(100)
      K1=K-1
      DO 2 I=1,K1
      S=0
      DO 1 J=1,K1
    1 S=S+A(K,I)*D(J,I)
    2 Y(I)=-BT*S
      RETURN
      END

      SUBROUTINE TB (N,X,Y,K,BT,B)
      DIMENSION B(100,100),X(100),Y(100)
      K1=K-1
      DO 1 I=1,K1
      DO 1 J=1,K1
    1 B(I,J)=X(I)*Y(J)/BT
      RETURN
      END


C     ***********************************
C     * P R O G R A M U L  I N V M A E *
C     ***********************************
C     INVERSAREA MATRICELOR PATRATICE PRIN METODA ELIMINARII
C
C
C   LISTA NOTATIILOR
C     N-DIMENSIUNEA MATRICEI CARE SE INVERSEAZA
C     B-INVERSA MATRICEI A
C     L-VECTORUL CARE INREGISTREAZA SCHIMBARILE DE LINII
C         CE AU LOC IN TIMPUL CALCULELOR
C
C   ALOCAREA MEMORIEI
      DIMENSION A(100,100),B(200,200)
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N
    1 FORMAT (I5)
      DO 2 I=1,N
    2 READ (1,3) (A(I,J),J=1,N)
    3 FORMAT (10F7.3)
      WRITE (3,4)
    4 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  I N V M A E*',
     *,/,24X,32('*'))
      WRITE (3,5)
    5 FORMAT (//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,6)
    6 FORMAT (/,35X,'MATRICEA A',/,35X,10('-'))
      DO 7 I=1,N
    7 WRITE (3,8) (A(I,J),J=1,N)
    8 FORMAT (/,13X,8F7.3)
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,9)
    9 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      CALL INVMAT (A,N,B)
      IF (B(N+1,1).EQ.0) GOTO 13
      WRITE (3,10)
   10 FORMAT (/,29X,'INVERSA B A MATRICEI A',/,29X,22('-'))
      DO 11 I=1,N
   11 WRITE (3,12) (B(I,J) J=1,N)
   12 FORMAT (/,5X,8(F7.3,1X))
      GOTO 15
   13 WRITE (3,14)
   14 FORMAT (//,25X.'MATRICEA A NU ESTE INVERSABILA')
   15 STOP
      END

      SUBROUTINE INVMAT(B,N,A)
      DIMENSION A(200.200),B(100,100),L(100)
      N2=N+2
      DO 1 I=1,N
      DO 1 J=1,N
      A(I,J)=B(I,J)
    1 A(I,J+N)=0
      DO 2 I=1,N
      A(I,N+I)=1
    2 L(I)=I
      DO 6 I=1,N
      IF (A(I,I).EQ.0) GOTO 8
    3 X=A(I,I)
      DO 4 K=1,N2
    4 A(I,K)=A(I,K)/X
      DO 6 K=1,N
      IF (K.EQ.I) GOTO 6
      X=A(K,I)
      IF (X.EQ.0) GOTO 6
      DO 5 J=1,N2
    5 A(K,J)=A(K,J)-A(I,J)*X
    6 CONTINUE
      DO 7 I=1,N
      DO 7 J=1,N
    7 A(L(I),J)=A(I,J+N)
      A(N+1,1)=1
      RETURN
    8 K=I+1
    9 IF (K.GT.N) GOTO 10
      IF (A(I,K).NE.0) GOTO 11
      K=K+1
      GOTO 9
   10 A(N+1,1)=0
      RETURN
   11 L1=L(I)
      L(I)=L(K)
      L(K)=L1
      DO 12 J=1,N
      R=A(K,N+J)
      A(K,N+J)=A(I,N+J)
      A(I,N+J)=R
      R=A(J,K)
      A(J,K)=A(J,I)
   12 A(J,I)=R
      GOTO 3
      END


C     ***********************************
C     * P R O G R A M U L  S I L G A U *
C     ***********************************
C     REZOLVAREA UNUI SISTEM DE N ECUATII LINIARE CU N NECUNOSCUTE
C                 PRIN METODA ELIMINARII GAUSS
C
C
C   LISTA NOTATIILOR
C     N-NUMARUL DE ECUATII SI NECUNOSCUTE DIN SISTEM
C     A-MATRICEA COEFICIENTILOR NECUNOSCUTELOR
C     B-VECTORUL TERMENILOR LIBERI
C     X-VECTORUL NECUNOSCUTELOR
C
C   ALOCAREA MEMORIEI
      DIMENSION A(100,100),W(100,100),B(100),X(100)
C
```


Scanned with OKEN Scanner

```
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N
    1 FORMAT (I5)
      DO 2 I=1,N
    2 READ (1,3) (A(I,J),J=1,N)
    3 FORMAT (6F10.5)
      READ (1,3) (B(I),I=1,N)
      WRITE (3,4)
    4 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  S I L G A U*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,5)
    5 FORMAT (/,33X,'MATRICEA A(I,J)',/,33X,15('*'))
      DO 6 I=1,N
    6 WRITE (3,7) (A(I,J),J=1,N)
    7 FORMAT (/,6(1X,F10.5))
      WRITE (3,8)
    8 FORMAT (/,34X,'VECTORUL B(I)',/,34X,13('*'))
      WRITE (3,7) (B(I),I=1,N)
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,9)
    9 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      DO 10 I=1,N
      DO 10 J=1,N
   10 W(I,J)=A(I,J)
      NN=N-1
      DO 17 I=1,NN
      N1=I
      DO 11 L=1,N
      IF (ABS(A(N1,I)).GE.ABS(A(L,I))) GOTO 11
      N1=L
   11 CONTINUE
      IF (A(N1,I)).EQ.0) GOTO 27
      IF (N1.EQ.I) GOTO 13
      DO 12 J=1,N
      R=A(I,J)
      A(I,J)=A(N1,J)
   12 A(N1,J)=R
      R=B(I)
      B(I)=B(N1)
      B(N1)=R
   13 R=A(I,I)
      DO 14 J=1,N
   14 A(I,J)=A(I,J)/R
      B(I)=B(I)/R
      N1=I+1
      DO 16 J=N1,N
      IF (A(J,I).EQ.0) GOTO 15
      R=A(J,I)
      DO 15 K=1,N
      A(J,K)=A(J,K)-R*A(I,K)
   15 CONTINUE
      B(J)=B(J)-B(I)*R
   16 CONTINUE
   17 CONTINUE
      IF (A(N,N).EQ.0) GOTO 27
      X(N)=B(N)/A(N,N)
      DO 19 I=1,NN
      I1=N-1
      X(I1)=B(I1)
      DO 18 K=1,I
      I2=I1+K
   18 X(I1)=X(I1)-A(I1,I2)*X(I2)
   19 CONTINUE
      WRITE (3,20)
   20 FORMAT (//,31X,'SOLUTIA SISTEMULUI',/,31X,18('-'),//,
     *31X,20('-'),/,32X,'I',11X,'X(I)',/,31X,20('-'))
      DO 23 I=1,N
      WRITE (3,21) I,X(I)
   21 FORMAT (/,31X,I2,4X,F12.6)
      IF (I.NE.N) GOTO 23
      WRITE (3,22)
   22 FORMAT (31X,20('-'),//)
   23 CONTINUE
      DO 25 I=1,N
      S=0.0
      DO 24 J=1,N
   24 S=S+W(I,J)*X(J)
   25 WRITE (3,26) I,S
   26 FORMAT (/,21X,'ECUATIA ',I2,' VERIFICARE ',E15.7)
      GOTO 29
   27 WRITE (3,28)
   28 FORMAT (/,22X,'SISTEM INCOMPATIBIL SAU NEDETERMINAT')
   29 STOP
      END
```

```
C     ********************************
C     *P R O G R A M U L  S I L J A C *
C     ********************************
C     REZOLVAREA SISTEMELOR DE ECUATII LINIARE CU METODA JACOBI
C
C
C   LISTA NOTATIILOR
C     N-NUMARUL DE ECUATII SI NECUNOSCUTE DIN SISTEM
C     NI-NUMARUL MAXIM DE ITERATII
C     EPS-EROAREA ADMISA IN CALCUL
C     A-MATRICEA COEFICIENTILOR NECUNOSCUTELOR
C     B-VECTORUL TERMENILOR LIBERI
C     X-VECTORUL CONTININD SOLUTIA INITIALA SI APOI FINALA
C     KOD-IDENTIFICATOR CU VALOAREA 0 SAU 1 DUPA CUM SOLUTIA
C        SE POATE OBTINE SAU NU IN MAXIMUM NI ITERATII
C
C   ALOCAREA MEMORIEI
      DIMENSION A(100,100),B(100),X(100),Y(100),Z(100),R(100)
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,NI,EPS
    1 FORMAT (2I5,F8.5)
      DO 2 I=1,N
    2 READ (1,3) (A(I,J),J=1,N)
    3 FORMAT (6F10.5)
      READ (1,3) (B(I),I=1,N)
      READ (1,3) (X(I),I=1,N)
      WRITE (3,4)
    4 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  S I L J A C*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,6) N,NI,EPS
    6 FORMAT (/,30X,'N=',I3,3X,'NI=',I3,3X,'EPS=',F7.5)
      WRITE (3,7)
    7 FORMAT (/,33X,'MATRICEA A(I,J)',/,33X,15('-'))
      DO 8 I=1,N
    8 WRITE (3,9) (A(I,J),J=1,N)
    9 FORMAT (/,6(1X,F10.5))
      WRITE (3,10)
   10 FORMAT (/,31X,'TERMENUL LIBER B(I)',/,31X,19('-'))
      WRITE (3,9) (B(I),I=1,N)
      WRITE (3,11)
   11 FORMAT (/,30X,'SOLUTIA INITIALA X(I)',/,30X,21('-'))
      WRITE (3,9) (X(I),I=1,N)
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,12)
   12 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      DO 14 I=1,N
      C=0
      DO 13 J=1,N
      IF (J.EQ.I) GOTO 13
      C=C+ABS(A(I,J)/A(I,I))
   13 CONTINUE
      IF (C.GE.1) GOTO 31
   14 CONTINUE
      DO 15 I=1,N
   15 Y(I)=X(I)
      K=1
   16 DO 19 I=1,N
      S=0
      DO 18 J=1,N
      IF (I-J) 17,18,17
   17 S=S+A(I,J)*Y(J)
   18 CONTINUE
      X(I)=(B(I)-S)/A(I,I)
```


Scanned with OKEN Scanner

```
   19 Y(I)=X(I)
      DO 20 I=1,N
      Z(I)=0
      DO 20 J=1,N
   20 Z(I)=Z(I)+A(I,J)*X(J)
      DO 21 L=1,N
      R(L)=ABS(Z(L)-B(L))
      IF (R(L)-EPS) 21,21,22
   21 CONTINUE
      KOD=0
      GOTO 25
   22 IF (K-NI) 23,23,24
   23 K=K+1
      GOTO 16
   24 KOD=1
   25 WRITE (3,26) KOD,K
   26 FORMAT (/,31X,'KOD=',I2,3X,'ITER=',I3)
      WRITE (3,27)
   27 FORMAT (/,31X,'SOLUTIA SISTEMULUI',/,31X,18('-'),//,
     *31X,20('-'),32X,'I',11X,'X(I)',/,31X,20('-'))
      DO 30 I=1,N
      WRITE (3,28) I,X(I)
   28 FORMAT (/,31X,I2,4X,F12.6)
      IF (I.NE.N) GOTO 30
      WRITE (3,29)
   29 FORMAT (31X,20('-'))
   30 CONTINUE
      GOTO 33
   31 WRITE (3,32)
   32 FORMAT (/,29X,'ALGORITM NECONVERGENT')
   33 STOP
      END
```

```
C     ********************************
C     * P R O G R A M U L  S I L G A S *
C     ********************************
C     REZOLVAREA SISTEMELOR DE ECUATII LINIARE CU METODA GAUSS-SEIDEL
C
C
C    LISTA NOTATIILOR
C      N-NUMARUL DE ECUATII SI NECUNOSCUTE DIN SISTEM
C      NI-NUMARUL MAXIM DE ITERATII
C      EPS-EROAREA ADMISA
C      A-MATRICEA COEFICIENTILOR NECUNOSCUTELOR
C      B-VECTORUL TERMENILOR LIBERI
C      X-VECTORUL CONTININD SOLUTIA INITIALA SI APOI FINALA
C      KOD-IDENTIFICATOR CU VALOAREA 0 SAU 1 DUPA CUM SOLUTIA
C         SE POATE OBTINE SAU NU IN MAXIMUM DE ITERATII
C
C    ALOCAREA MEMORIEI
      DIMENSION A(100,100),B(100),X(100),Y(100),R(100)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,NI,EPS
    1 FORMAT (2I5,F8.5)
      DO 2 I=1,N
    2 READ (1,3) (A(I,J),J=1,N)
    3 FORMAT (6F10.5)
      READ (1,3) (B(I),I=1,N)
      READ (1,3) (X(I),I=1,N)
      WRITE (3,4)
    4 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  S I L G A S*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,6) N,NI,EPS
    6 FORMAT (/,30X,'N=',I3,3X,'NI=',I3,3X,'EPS=',F7.5)
      WRITE (3,7)
    7 FORMAT (/,33X,'MATRICEA A(I,J)',/,33X,15('-'))
      DO 8 I=1,N
    8 WRITE (3,9) (A(I,J),J=1,N)
    9 FORMAT (/,6(1X,F10.5))
      WRITE (3,10)
   10 FORMAT (/,31X,'TERMENUL LIBER B(I)',/,31X,19('-'))
      WRITE (3,9) (B(I),I=1,N)
      WRITE (3,11)
   11 FORMAT (/,30X,'SOLUTIA INITIALA X(I)',/,30X,21('-'))
      WRITE (3,9) (X(I),I=1,N)
C
C    EFECTUAREA CALCULULUI SI SCRIEREA REZULTATELOR
      WRITE (3,12)
   12 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      DO 14 I=1,N
      C=0
      DO 13 J=1,N
      IF (J.EQ.I) GOTO 13
      C=C+ABS(A(I,J)/A(I,I))
   13 CONTINUE
      IF (C.GE.1) GOTO 30
   14 CONTINUE
      DO 15 I=1,N
   15 Y(I)=X(I)
      WRITE (3,16)
   16 FORMAT (/,31X,'SOLUTIA SISTEMULUI',/,31X,16('-'),//)
      DO 27 K=1,NI
      S=0
      DO 17 I=2,N
   17 S=S+A(1,I)*X(I)
      X(1)=(B(1)-S)/A(1,1)
      Y(1)=X(1)
      DO 24 I=2,N
      I1=I-1
      I2=I+1
      S1=0
      S2=0
      DO 18 J=1,I1
   18 S1=S1+A(I,J)*Y(J)
      IF (I2-N) 19,19,23
   19 DO 20 J=I2,N
   20 S2=S2+A(I,J)*Y(J)
      IF (A(I,I)) 23,21,23
   21 WRITE (3,22) I,I
   22 FORMAT (/,35X,'A(',I2,I2,')=0',/)
      GOTO 32
   23 X(I)=(B(I)-S1-S2)/A(I,I)
      R(I)=ABS(X(I)-Y(I))
   24 Y(I)=X(I)
      WRITE (3,25) (X(L),L=1,N)
   25 FORMAT (/,5(1X,F12.6),/)
      DO 26 I=1,N
      IF (R(I)-EPS) 26,26,27
   26 CONTINUE
      KOD=0
      GOTO 28
   27 CONTINUE
      KOD=1
   28 WRITE (3,29) KOD,K
   29 FORMAT (/,31X,'KOD=',I2,3X,'ITER=',I3)
      GOTO 32
   30 WRITE (3,31)
   31 FORMAT (/,29X,'ALGORITM NECONVERGENT')
   32 STOP
      END
```

```
C     ********************************
C     * P R O G R A M U L  P C K R I L *
C     ********************************
C     DETERMINAREA COEFICIENTILOR POLINOMULUI CARACTERISTIC ASOCIAT
C                  UNEI MATRICE CU METODA KRILOV
C
C
C    LISTA NOTATIILOR
C      N-DIMENSIUNEA MATRICEI DATE A
C      Y-VECTOR ALES INITIAL
C      T-VECTOR DE LUCRU
C      V-MATRICE DE LUCRU
C      TETA-VECTOR CE CONTINE COEFICIENTII POLINOMULUI IN ORDINEA
C           DESCRESCATOARE A PUTERILOR
C
C    ALOCAREA MEMORIEI
      DIMENSION A(100,100),Y(100),V(100,100),T(100),TETA(100)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
```


Scanned with OKEN Scanner

```
      READ (1,1) N
    1 FORMAT (I3)
      DO 2 I=1,N
    2 READ (1,3) (A(I,J),J=1,N)
    3 FORMAT (6F10.5)
      READ (1,3) (Y(I),I=1,N)
      WRITE (3,4)
    4 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  P C K R I L*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,6)
    6 FORMAT (/,33X,'MATRICEA A(I,J)',/,33X,15('-'))
      DO 7 I=1,N
    7 WRITE (3,8) (A(I,J),J=1,N)
    8 FORMAT (/,6(1X,F10.5))
      WRITE (3,9)
    9 FORMAT (/,29X,'VECTORUL INITIAL Y(I)',/,29X,21('-'))
      WRITE (3,8) (Y(I),I=1,N)
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,10)
   10 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      DO 15 I=1,N
      K=N-I+1
      DO 11 J=1,N
   11 V(J,K)=Y(J)
      DO 13 L=1,N
      T(L)=0
      DO 12 J=1,N
   12 T(L)=T(L)+A(I,J)*Y(J)
   13 CONTINUE
      DO 14 K=1,N
   14 Y(K)=T(K)
   15 CONTINUE
      WRITE (3,16)
   16 FORMAT (/,27X,'MATRICEA SISTEMULUI V(I,J)',/,27X,26('-'))
      DO 17 I=1,N
   17 WRITE (3,18) (V(I,J),J=1,N)
   18 FORMAT (/,10X,6F9.3)
      WRITE (3,19)
   19 FORMAT (/ 24X,'VECTORUL TERMENILOR LIBERI T(I)',/,24X,31('-'))
      WRITE (3,18) (T(I),I=1,N)
      NN=N-1
      DO 26 I=1,NN
      N1=I
      DO 20 L=1,N
      IF (ABS(V(N1,I)).GE.ABS(V(L,I))) GOTO 20
      N1=L
   20 CONTINUE
      IF (V(N1,I).EQ.0) GOTO 30
      IF (N1.EQ.I) GOTO 22
      DO 21 J=1,N
      R=V(I,J)
      V(I,J)=V(N1,J)
      V(N1,J)=R
   21 CONTINUE
      R=T(I)
      T(I)=T(N1)
      T(N1)=R
   22 R=V(I,I)
      DO 23 J=1,N
   23 V(I,J)=V(I,J)/R
      T(I)=T(I)/R
      N1=I+1
      DO 25 J=N1,N
      IF (V(J,I).EQ.0) GOTO 24
      R=V(J,I)
      DO 24 K=1,N
      V(J,K)=V(J,K)-R*V(I,K)
   24 CONTINUE
      T(J)=T(J)-T(I)*R
   25 CONTINUE
   26 CONTINUE
      IF (V(N,N).EQ.0) GOTO 30
      TETA(N)=T(N)/V(N,N)
      DO 28 I=1,NN
      I1=N-I
      TETA(I1)=T(I1)
      DO 27 K=1,I
      I2=I1+K
   27 TETA(I1)=TETA(I1)-V(I1,I2)*TETA(I2)
   28 CONTINUE
      WRITE (3,29)
   29 FORMAT (/,21X,'COEFICIENTII POLINOMULUI CARACTERISTIC',/,21X,38('-'))
      WRITE (3,18) (TETA(I),I=1,N)
      GOTO 32
   30 WRITE (3,31)
   31 FORMAT (/,22X,'SISTEM INCOMPATIBIL SAU NEDETERMINAT')
   32 STOP
      END
```

```
C     ********************************
C     * P R O G R A M U L  S I N N E R *
C     ********************************
C     REZOLVAREA SISTEMELOR DE ECUATII NELINIARE CU METODA NEWTON-RAPHSON
C
C
C   LISTA NOTATIILOR
C     N-NUMARUL DE ECUATII SI NECUNOSCUTE DIN SISTEM
C     NI-NUMARUL MAXIM DE ITERATII
C     EPS-EROAREA ADMISA
C     X-VECTOR CONTININD APROXIMAREA SOLUTIEI SI APOI SOLUTIA SISTEMULUI
C     B-VECTOR CONTININD VALORILE FUNCTIEI MATRICEALE F(X)
C     AJ-MATRICEA JACOBIAN A SISTEMULUI
C     KOD-IDENTIFICATOR CU VALOAREA 0 SAU 1 DUPA CUM SOLUTIA
C         SE POATE OBTINE SAU NU IN MAXIMUM DE ITERATII
C
C   ALOCAREA MEMORIEI
      DIMENSION AJ(100,100),B(100),X(100)
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,NI,EPS
    1 FORMAT (2I5,F8.5)
      READ (1,2) (X(I),I=1,N)
    2 FORMAT (6F10.5)
      WRITE (3,3)
    3 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  S I N N E R*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,5) N,NI,EPS
    5 FORMAT (/,30X,'N=',I3,3X,'NI=',I3,3X,'EPS=',F7.5)
      WRITE (3,6)
    6 FORMAT (/,27X,'APROXIMATIA INITIALA X0(I)',/,27X,26('-'))
      WRITE (3,7) (X(I),I=1,N)
    7 FORMAT (/,6(1X,F10.5))
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,8)
    8 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      K=1
    9 AJ(1,1)=1
      AJ(1,2)=1
      AJ(1,3)=0
      AJ(2,1)=1
      AJ(2,2)=-1/X(2)
      AJ(2,3)=1
      AJ(3,1)=0
      AJ(3,2)=2*X(2)
      AJ(3,3)=-2
      B(1)=-X(1)-X(2)+2
      B(2)=-X(1)+2.30258*ALOG10 (X(2))-X(3)+2
      B(3)=-X(2)**2+2*X(3)-1
      M=0
      DO 10 I=1,N
      IF (ABS(B(I)).GE.EPS) M=1
   10 CONTINUE
      IF (M.NE.0) GOTO 11
      KOD=1
      GOTO 13
   11 CALL SISTLIN (N,EPS,AJ,B)
      DO 12 I=1,N
   12 X(I)=X(I)+B(I)
      K=K+1
      IF (K.LT.NI) GOTO 9
```


Scanned with OKEN Scanner

```
      KOD=2
   13 WRITE (3,14) KOD,K
   14 FORMAT (/,31X,'KOD=',I2,3X,'ITER=',I3)
      WRITE (3,15)
   15 FORMAT (/,29X,'SOLUTIA SISTEMULUI X(I)',/,29X,23('-'))
      WRITE (3,16) (X(I),I=1,N)
   16 FORMAT (/,5(1X,F12.6))
      STOP
      END

      SUBROUTINE SISTLIN (N,EPS,A,B)
      DIMENSION A(100,100),B(100)
      K1=N-1
      DO 5 K=1,K1
      AM=0
      DO 1 I=K,N
      IF (ABS(A(I,K)).LE.AM) GOTO 1
      AM=ABS(A(I,K))
      IO=I
    1 CONTINUE
      IF (AM.LE.EPS) GOTO 8
      DO 2 J=K,N
      X=A(K,J)
      A(K,J)=A(IO,J)
    2 A(IO,J)=X
      X=B(K)
      B(K)=B(IO)
      B(IO)=X
      K2=K+1
      DO 4 I=K2,N
      X=A(I,K)/A(K,K)
      DO 3 J=K,N
    3 A(I,J)=A(I,J)-A(K,J)*X
    4 B(I)=B(I)-B(K)*X
    5 CONTINUE
      IF (ABS(A(N,N)).LT.EPS) GOTO 8
      B(N)=B(N)/A(N,N)
      K=N-1
    6 K1=K+1
      DO 7 J=K1,N
    7 B(K)=B(K)-A(K,J)*B(J)
      B(K)=B(K)/A(K,K)
      K=K-1
      IF (K.GE.1) GOTO 6
      GOTO 10
    8 WRITE (3,9)
    9 FORMAT (/,27X,'SISTEM LINIAR INCOMPATIBIL')
      STOP
   10 RETURN
      END


C     ********************************
C     * P R O G R A M U L  S I N S U R *
C     ********************************
C     REZOLVAREA SISTEMELOR DE ECUATII NELINIARE CU METODA
C               SUPRARELAXARILOR SUCCESIVE
C
C
C   LISTA NOTATIILOR
C     N-NUMARUL DE ECUATII SI NECUNOSCUTE DIN SISTEM
C     NI-NUMARUL MAXIM DE ITERATII
C     EPS-EROAREA ADMISA
C     OMEGA-FACTOR DE RELAXARE
C     H-CONSTANTA POZITIVA
C     X-VECTOR CONTININD APROXIMAREA SOLUTIEI SI APOI SOLUTIA SISTEMULUI
C
C   ALOCAREA MEMORIEI
C     DIMENSION X(200)
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,NI,EPS,OMEGA,H
    1 FORMAT (2I5,F8.5,2F6.3)
      READ (1,2) (X(I),I=1,N)
    2 FORMAT (6F10.5)
      WRITE (3,3)
    3 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  S I N S U R*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,5) N,NI EPS,OMEGA,H
    5 FORMAT (/,10X,'N=',I3,3X,'NI=',I3,3X,'EPS=',F7.5,3X,
     *'OMEGA=',F4.2,3X,'H=',F5.2)
      WRITE (3,6)
    6 FORMAT (/,27X,'APROXIMATIA INITIALA XO(I)',/,27X,26('-'))
      WRITE (3,7) (X(I),I=1,N)
    7 FORMAT (/,6(1X,F10.5))
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,8)
    8 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      K=1
    9 J=0
      DO 12 I=1,N
      Y=X(I)
      CALL VALF (N,I,X,F)
      X(I)=X(I)+H
   10 CALL VALF (N,I,X,F1)
      XI=X(I)
      X(I)=X(I)-H-H*F/(F1-F)
      IF (ABS(F1).LE.EPS) GOTO 11
      F=F1
      H=X(I)-XI
      GOTO 10
   11 X(I)=Y+OMEGA*(X(I)-Y)
      IF (ABS(X(I)-Y).GT.EPS) J=1
   12 CONTINUE
      IF (J.NE.0) GOTO 13
      KOD=1
      GOTO 14
   13 K=K+1
      IF (K.LE.NI) GOTO 9
      KOD=2
   14 WRITE (3,15) KOD,K
   15 FORMAT (/,31X,'KOD=',I2,3X,'ITER=',I3)
      WRITE (3,16)
   16 FORMAT (/,29X,'SOLUTIA SISTEMULUI X(I)',/,29X,23('-'))
      WRITE (3,17) (X(I),I=1,N)
   17 FORMAT (/,5(1X,F12.6))
      STOP


      END


      SUBROUTINE VALF (N,I,X,F)
      DIMENSION X(200)
      GOTO (1,2,3),I
    1 F=X(1)+X(2)-2
      RETURN
    2 F=X(1)-2,30258*ALOG10 (X(2))-2
      RETURN
    3 F=X(2)**2-X(3)
      RETURN
      END


C     ************************************
C     * P R O G R A M U L  E V G H E C A *
C     ************************************
C     ANALIZA NUMERICA A EVOLUTIEI IN TIMP A GHETII IN CONDUCTELE
C                     AERIENE SUB PRESIUNE
C
C
C   LISTA NOTATIILOR
C     CI-PANTA CONDUCTEI
C     G-DEBITUL DE APA IN CONDUCTA IN MC/S
C     RO-RAZA INTERIOARA A CONDUCTEI IN M
C     RC-RAZA DE CALCUL IN M
C     TE-TEMPERATURA AERULUI EXTERIOR IN GRD
C     W-VITEZA VINTULUI IN M
C     DT-PASUL DE TIMP IN ZILE
C     TAU(I)-TIMPUL IN ZILE
C     R(I)-RAZA RELATIVA A GHETII (RO/RC)
```


Scanned with OKEN Scanner

```
C      RLIM-RAZA RELATIVA LIMITA CATRE CARE TINDE GHEATA DIN CONDUCTA
C          DUPA TIMPUL TAULIM
C
C    ALOCAREA MEMORIEI
       DIMENSION R(500),TAU(500)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
       READ (1,1) G,CI,RO,TE W,DT,EPS1,EPS2
     1 FORMAT (F6.3,F7.4,F5.2,F6.1,F5.2,2F6.5)
       WRITE (3,2)
     2 FORMAT (////,23X,34('*'),/,23X,'*P R O G R A M U L  E V G H E C A*',
      */,23X,34('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
       WRITE (3,3) G,CI,RO.TE,W,DT,EPS1,EPS2
     3 FORMAT (/,10X,'G=',F6.3,' NC/S',2X,'CI=',F6.4,2X,'RO=',F5.2,' M',
      *2X,'TE=',F5.1,' GRD',/,10X,'W=',F4.1, 1.70141E+38 ,2X,'DT=',F4.2,' ZILE',
      *2X,'EPS1=',F6.4,2X,'EPS2=',F7.5)
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
       WRITE (3,4)
     4 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
       RC=RO*EXP(0.615/(W*RO)**0.7)
       A=0.148*G*CI/RC**2
       B=7.48/10**6*G**3/RC**7.33
       C=-0.66/10**3*TE/RC**2
       I=1
       TAU(I)=0
       R(I)=RO/RC
     5 X=R(I)
       CALL FUNCT (A,B,C,X,F)
       S1=DT*F
       X=R(I)+S1/2
       CALL FUNCT (A,B,C,X,F)
       S2=DT*F
       X=R(I)+S2/2
       CALL FUNCT (A,B,C,X,F)
       S3=DT*F
       IF (ABS(S3-S2).LE.EPS1) GOTO 6
       DT=DT/2
       GOTO 5
     6 X=R(I)+S3
       CALL FUNCT (A,B,C,X,F)
       S4=DT*F
       TAU(I+1)=TAU(I)+DT
       R(I+1)=R(I)+(S1+2*S2+2*S3+S4)/6
       N=I+1
       IF (ABS(R(N)-R(I)).LE.EPS2) GOTO 7
       I=I+1
       GOTO 1
     7 RLIM=R(N)
       TAULIM=TAU(N)
       WRITE (3,8) RLIM,TAULIM
     8 FORMAT (//,20X,'RLIM=',F6.4,3X,'TAULIM=',F5.1,' ZILE',//,
      *29X,22('-'),/,30X,'I',5X,'TAU',7X,'R',/,29X,22('-'))
       DO 11 I=1,N
       WRITE (3,9) I,TAU(I),R(I)
     9 FORMAT (/,29X,I3,3X,F5.1,3X,F8.5)
       IF (I.NE.N) GOTO 11
       WRITE (3,10)
    10 FORMAT (29X,22('-'))
    11 CONTINUE
       RAD=0.705*RO/RC
       IF (RLIM.GE.RAD) GOTO 13
       WRITE (3,12)
    12 FORMAT (//,20X,'ESTE NECESARA IZOLAREA CONDUCTEI')
    13 TAUMAX=758*RO**2/TE*(0.503*ALOG(RO/RC)-0.076)
       WRITE (3,14) TAUMAX
    14 FORMAT (//,7X,'TIMPUL ADMISIBIL DE STAGNARE A APEI',
      *'  TAUMAX',F6.2,' ZILE')
       STOP
       END

       SUBROUTINE FUNCT (A,B,C,X,F)
       F=A/X+B/X**6.33+C/(X*ALOG(X))
       RETURN
       END
```

```
C      ********************************
C      * P R O G R A M U L  A T E F S 1 *
C      ********************************
C      ANALIZA TERMICA CU ELEMENTE FINITE LINIARE IN REGIM STATIONAR
C                PENTRU PROBLEME PLANE AXIAL SIMETRICE
C
C
C    ALOCAREA MEMORIEI
       DIMENSION S(32),S1(32)
       COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
       COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
       COMMON/BL3/ICON(400),ID(450,3),TETA(400)
       COMMON/BL4/SE(3,3),FE(3)
       COMMON/BL5/A(3),B(3),C(3),DELTA,DET
       COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
C
C    CITESTE DATELE DE INTRARE DE PE FLOPPY-DISC SI CONSTITUIE
C    MATRICELE DE CONDUCTIVITATE SI VECTORUL INCARCARII TERMICE
       NOD=3
       CALL INN(NFLAG)
       DO 3 I=1,NN
       X(I)=X(I)/100.
     3 Y(I)=Y(I)/100.
       DO 5 I=1,32
       S1(I)=I
     5 S(I)=0
       CALL OPEN(6,'MATRICEAS  ',2)
       I=1
    10 WRITE (6,REC=I) (S(K),K=1,32)
       GOTO 15
    12 WRITE (6,REC=I) (S1(K),K=1,32)
    15 I=I+1
       IF (I.LE.NSIZE) GOTO 10
       IF (I.LT.NSIZE+31) GOTO 12
       WRITE (6,REC=I) (S1(K),K=1,32)
       REWIND 6
       CALL FMK(NFLAG)
       ENDFILE 6
       CALL OPEN (5,'NSIZEMBAND ',2)
       WRITE (5,REC=1) NSIZE,MBAND,NN
       ENDFILE 5
       STOP
       END


       SUBROUTINE INN (NFLAG)
C      CITESTE DATELE DE INRARE DE PE FLOPPY-DISK
       DIMENSION ITITL(80)
       COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
       COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
       COMMON/BL3/ICON(400),ID(450,3),TETA(400)
       COMMON/BL4/SE(3,3),FE(3)
       COMMON/BL5/A(3),B(3),C(3),DELTA,DET
       COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
       READ (6,500) NPROBL,NFLAG,NE,NN,NNTIMP,NECONV,NEFLUX
       READ (6,550) (ITITL(K),K=1,80)
       ENDFILE 6
       DO 2 K=1,NN
       READ (7,650) KI,X(K),Y(K),TETA(K),ICON(K)
     2 CONTINUE
       ENDFILE 7
       DO 4 K=1,NE
       READ (8,700) KI,ID(K,1),ID(K,2),ID(K,3),FLINT(K),ALAMDX(K),ALAMDY(K)
     4 CONTINUE
       ENDFILE 8
       IF (NEFLUX.EQ.0) GOTO 8
       DO 6 K=1,NEFLUX
       READ (9,750) KI,NFLUX(K,1),NFLUX(K,2),NFLUX(K,3),FLU(K)
     6 CONTINUE
       ENDFILE 9
     8 CONTINUE
       IF (NECONV.EQ.0) GOTO 12
       DO 10 K=1,NECONV
       READ (10,800) KI,NCON(K,1),NKON(K,2),NCON(K,3),ALFA(K),TETAE(K)
    10 CONTINUE
       ENDFILE 10
```


Scanned with OKEN Scanner

```
   12 CONTINUE
      WRITE (3,850)
   14 READ (4,900) NC
      IF (NC.EQ.0) GOTO 100
      IF (NC.EQ.1) NWR=3
      IF (NC.EQ.2) NWR=2
      WRITE (NWR,1000) NPROBL
      WRITE (NWR,1100) (ITITL(K),K=1,70)
      WRITE (NWR,1200)
      WRITE (NWR,1300) NE,NN,NNTIMP,NECONV,NEFLUX
      WRITE (NWR,1400)
      DO 20 I=1,NN
      WRITE (NWR,1500) I,X(I),Y(I),TETA(I),ICON(I)
   20 CONTINUE
      WRITE (NWR,1600)
      DO 40 I=1,NE
      WRITE (NWR,1700) I,(ID(I,K),K=1,3),FLINT(I),ALAMDX(I),ALAMDY(I)
   40 CONTINUE
      IF (NEFLUX.EQ.0) GOTO 70
      WRITE (NWR,1800)
      DO 60 I=1,NEFLUX
      WRITE (NWR,1900) (NFLUX(I,K),K=1,3),FLU(I)
   60 CONTINUE
   70 CONTINUE
      IF (NECONV.EQ.0) GOTO 100
      WRITE (NWR,2000)
      DO 90 I=1,NECONV
      WRITE (NWR,1900) (NCON(I,K),K=1,3),ALFA(I),TETAE(I)
   90 CONTINUE
  100 CONTINUE
      CALL BAD
  500 FORMAT (7I5)
  550 FORMAT (80A1)
  650 FORMAT (I5,3F10.2,I5)
  700 FORMAT (4I5,F10.0,2F10.2)
  750 FORMAT (4I5,F10.0)
  800 FORMAT (4I5,2F10.2)
  850 FORMAT (10X,'INTRODUCETI CIFRA',/
     *2X,'0,DACA NU DORITI DATELE DE INTRARE',/
     *2X,'1,DACA DORITI DATELE DE INTRARE AFISATE PE ECRAN',/
     *2X,'2,DACA DORITI DATELE DE INTRARE IMPRIMATE',/)
  900 FORMAT (I5)
 1000 FORMAT (//,10X,'PROBLEMA NR. ',I2,///)
 1100 FORMAT (2X,70A1,//)
 1200 FORMAT (10X,' NE  NN  NNTIMP  NECONV  NEFLUX',/)
 1300 FORMAT (9X,2(1X,I3),3(5X,I3),//)
 1400 FORMAT (//,25X,'DATE NODALE',//,10X,
     *'N          X          Y        TETA       ICON',//)
 1500 FORMAT (7X,I5,3F10.2,4X,I5)
 1600 FORMAT (//,20X,'DATELE ELEMENTELOR',//,5X,
     *'N          ID          FLINT       ALAMDX       ALAMDY',//)
 1700 FORMAT (2X,4I5,F10.0,2F10.2)
 1800 FORMAT (//,10X,'DATE PT. FLUX TERMIC',//,10X,
     *'NFLUX          FLU',//)
 1900 FORMAT (2X,3I5,2F12.2)
 2000 FORMAT (//,15X,'DATE PT. CONVECTIE',//,10X,
     *'NCON          ALFA          TETAE',//)
      RETURN
      END



      SUBROUTINE BAD
C     CALCULEAZA NSIZE SI MBAND
      DIMENSION KB(3)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
      COMMON/BL3/ICON(400),ID(450,3),TETA(400)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
      NSIZE=0
      MBAND=0
      DO 140 II=1,NN
      IF (ICON(II).GT.0) GOTO 135
      NSIZE=NSIZE+1
      ICON(II)=NSIZE
      GOTO 140
  135 ICON(II)=0
  140 CONTINUE
      DO 160 N=1,NE
      DO 150 I=1,NOD
      IJ=ID(N,I)
  150 KB(I)=ICON(IJ)
      MAX=0
      MIN=10000
      DO 155 I=1,NOD
      IF (KB(I).EQ.0) GOTO 155
      IF (KB(I).GT.MAX) MAX=KB(I)
      IF (KB(I).LT.MIN) MIN=KB(I)
  155 CONTINUE
      NDIF=MAX-MIN+1
      IF (NDIF.GT.MBAND) MBAND=NDIF
  160 CONTINUE
      WRITE (2,200) NSIZE,MBAND
  200 FORMAT (//,5X,'NUMARUL DE ECUATII, NSIZE=',I4,/
     *          5X,'SEMILATIMEA BENZII, MBAND=',I4,/)
      RETURN
      END



      SUBROUTINE FMK (NFLAG)
C     CONSTITUIE MATRICELE DE CONDUCTIVITATE TERMICA ALE ELEMENTELOR SI
C     VECTORII INCARCARII TERMICE PE ELEMENTE
      DIMENSION RP$(3),ZP$(3)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
      COMMON/BL3/ICON(400),ID(450,3),TETA(400)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
      WRITE (3,900)
      DO 90 ME=1,NE
      WRITE (3,1000) ME
      DO 5 I=1,NOD
      FE(I)=0.
      DO 5 J=1,NOD
    5 SE(I,J)=0.
      CALL SHP (II,JJ,MM,ME)
C     CONSTITUIE MATRICELE SE PT. CONDITII LA LIMITA DE TIP 1.
      DO 10 I=1,NOD
      DO 10 J=1,NOD
   10 SE(I,J)=0.25*(B(I)*B(J)*ALAMDX(ME)+C(I)*C(J)*ALAMDY(ME))/DELTA
C     MODIFICARE SE DACA PROBLEMA E AXIAL SIMETRICA
      IF (NFLAG.NE.1) GOTO 12
      RAZA=(X(II)+X(JJ)+X(MM))/3.
      PI=3.1415927
      RAZA=2.*PI*RAZA*RAZA
      DO 11 I=1,NOD
      DO 11 J=1,NOD
   11 SE(I,J)=SE(I,J)*RAZA
   12 CONTINUE
C     VECTORUL TERMENULUI LIBER (INCARCARII),FE
      IF (FLINT(ME).EQ.0.) GOTO 13
      QQ$1=FLINT(ME)
      QQ$2=FLINT(ME)
      QQ$3=FLINT(ME)
      CONSTQ=DELTA/12
      FE(1)=(2.*QQ$1+QQ$2+QQ$3)*CONSTQ
      FE(2)=(QQ$1+2.*QQ$2+QQ$3)*CONSTQ
      FE(3)=(QQ$1+QQ$2+2.*QQ$3)*CONSTQ
C     MODIFICA VECTORUL FE DACA PROBLEMA E AXIAL SIMETRICA
      IF (NFLAG.NE.1) GOTO 13
      RBAR=(X(II)+X(JJ)+X(MM))/3.
      ZBAR=(Y(II)+Y(JJ)+Y(MM))/3.
      RP$(1)=RBAR-X(II)
      RP$(2)=RBAR-X(JJ)
      RP$(3)=RBAR-X(MM)
      ZP$(1)=ZBAR-Y(II)
      ZP$(2)=ZBAR-Y(JJ)
      ZP$(3)=ZBAR-Y(MM)
      FACT1=(RP$(1)**2+RP$(2)**2+RP$(3)**2)/12.
      FACT2=(RP$(1)*ZP$(1)+RP$(2)*ZP$(2)+RP$(3)*ZP$(3))/12.
```


Scanned with OKEN Scanner

```
      FACT3=RBAR*FLINT(ME)*PI
      FE(1)=FACT3*(A(1)*RBAR+B(1)*(RBAR**2)+B(1)*FACT1+C(1)*RBAR*ZBAR+
     *C(1)*FACT2)
      FE(2)=FACT3*(A(2)*RBAR+B(2)*(RBAR**2)+B(2)*FACT1+C(2)*RBAR*ZBAR+
     *C(2)*FACT2)
      FE(3)=FACT3*(A(3)*RBAR+B(3)*(RBAR**2)+B(3)*FACT1+C(3)*RBAR*ZBAR+
     *C(3)*FACT2)
   13 CONTINUE
C     COMPLETARE PENTRU SCHIMB DE CALDURA PE CONTUR PRIN CONVECTIE
      IF (NECONV.EQ.0) GOTO 20
      DO 15 I=1,NECONV
      IF (ME.NE.NCON(I,1)) GOTO 15
      NME=I
      CALL CON (NME,NFLAG)
      GOTO 20
   15 CONTINUE
   20 CONTINUE
C     COMPLETARE PENTRU SCHIMB PRIN FLUX TERMIC PE CONTUR
      IF (NEFLUX.EQ.0) GOTO 25
      DO 22 I=1,NEFLUX
      IF (ME.NE.NFLUX(I,1)) GOTO 22
      NME=I
      CALL FLX (NME,NFLAG)
      GOTO 25
   22 CONTINUE
   25 CONTINUE
C     MODIFICAREA SISTEMULUI DE ECUATII AL ELEMENTULUI DACA EXISTA
C     NODURI CU TEMPERATURA IMPUSA
      IF (NNTIMP.EQ.0) GOTO 80
      IF (ICON(II).NE.0) GOTO 40
      IF (TETA(II).EQ.0.) GOTO 40
      FE(2)=FE(2)-SE(2,1)*TETA(II)
      FE(3)=FE(3)-SE(3,1)*TETA(II)
   40 IF (ICON(JJ).NE.0) GOTO 45
      IF (TETA(JJ).EQ.0.) GOTO 45
      FE(1)=FE(1)-SE(1,2)*TETA(JJ)
      FE(3)=FE(3)-SE(3,2)*TETA(JJ)
   45 IF (ICON(MM).NE.0) GOTO 50
      IF (TETA(MM).EQ.0.) GOTO 50
      FE(1)=FE(1)-SE(1,3)*TETA(MM)
      FE(2)=FE(2)-SE(2,3)*TETA(MM)
   50 CONTINUE
   80 CONTINUE
C     ASAMBLAREA SISTEMULUI ELEMENTULUI CURENT IN SISTEMUL CENTRAL
      CALL ASM (II,JJ,MM)
   90 CONTINUE
  900 FORMAT (//,3X,'CONSTITUIREA ECUATIILOR ELEMENTALE',/)
 1000 FORMAT (1X,'ELEMENTUL',I4)
      RETURN
      END



      SUBROUTINE SHP (II,JJ,MM,ME)
      CALCULEAZA COEFICIENTII FUNCTIILOR DE FORMA SI ARIA ELEMENTULUI CURENT
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
      COMMON/BL3/ICON(400),ID(450,3),TETA(400)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
      II=ID(ME,1)
      JJ=ID(ME,2)
      MM=ID(ME,3)
      DET=(X(JJ)*Y(MM)-X(MM)*Y(JJ))-(X(II)*Y(MM)-X(MM)*Y(II))+
     *(X(II)*Y(JJ)-X(JJ)*Y(II))
      DELTA=0,5*DET
      A(1)=X(JJ)*Y(MM)-X(MM)*Y(JJ)
      B(1)=Y(JJ)-Y(MM)
      C(1)=X(MM)-X(JJ)
      A(2)=X(MM)*Y(II)-X(II)*Y(MM)
      B(2)=Y(MM)-Y(II)
      C(2)=X(II)-X(MM)
      A(3)=X(II)*Y(JJ)-X(JJ)*Y(II)
      B(3)=Y(II)-Y(JJ)
      C(3)=X(JJ)-X(II)
      RETURN
      END
```

```
      SUBROUTINE CON (NME,NFLAG)
C     ADAUGA LA SE SI FE CONTRIBUTIA TRANSFERULUI PRIN CONVECTIE PE CONTUR
      DIMENSION SEC(3,3),FEC(3)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
      COMMON/BL3/ICON(400),ID(450,3),TETA(400)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
C     NUMARUL ELEMENTULUI FINIT SI AL NODURILOR CU SCHIMB CONVECTIV
      NREL=NCON(NME,1)
      NODK=NCON(NME,2)
      NODL=NCON(NME,3)
      XK=X(NODK)
      YK=Y(NODK)
      XL=X(NODL)
      YL=Y(NODL)
      DX=XK-XL
      DY=YK-YL
C     LUNGIMEA LATURII CU SCHIMB CONVECTIV
      DLKL=SQRT(DX*DX+DY*DY)
      DO 10 I=1,3
      FEC(I)=0
      DO 10 J=1,3
   10 SEC(I,J)=0
      H=1
      IF (NFLEG.NE.1) GOTO 20
      PI=3.1415927
      H=PI*(XL+XK)
   20 CONTINUE
C     POZITIA NODULUI OPUS LATURII CU SCHIMB CONVECTIV IN MATRICEA ID
      DO 30 I=1,3
      IF (ID(NREL,I).EQ.NODK) GOTO 30
      IF (ID(NREL,I).EQ.NODL) GOTO 30
      NLC=I
      GOTO 32
   30 CONTINUE
C     MATRICELE PENTRU SCHIMB CONVECTIV
   32 DO 40 I=1,3
      IF (I.EQ.NLC) GOTO 40
      FEC(I)=0.5*H*ALFA(NME)*DLKL*TETAE(NME)
      DO 40 J=1,3
      IF (J.EQ.NLC) GOTO 40
      IF (I.NE.J) GOTO 35
      SEC(I,J)=H*DLKL*ALFA(NME)/3
      GOTO 40
   35 SEC(I,J)=H*DLKL*ALFA(NME)/6
   40 CONTINUE
C     INSUMEAZA EFECTUL CONVECTIEI LA MATRICELE SE SI FE
      DO 50 I=1,3
      FE(I)=FE(I)+FEC(I)
      DO 50 J=1,3
      SE(I,J)=SE(I,J)+SEC(I,J)
   50 CONTINUE
      RETURN
      END



      SUBROUTINE FLX (NME,NFLAG)
C     ADAUGA LA ELEMENTELE VECTORULUI TERMEN LIBER CONTRIBUTIA PRIN
C     FLUX TERMIC IMPUS PE CONTUR
      DIMENSION FEF(3)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
      COMMON/BL3/ICON(400),ID(450,3),TETA(400)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
C     NUMARUL ELEMENTULUI FINIT SI AL NODURILOR CU SCHIMB PRIN FLUX TERMIC
      NREL=NFLUX(NME,1)
      NODK=NFLUX(NME,2)
      NODL=NFLUX(NME,3)
      XK=X(NODK)
      YK=Y(NODK)
      XL=X(NODL)
      YL=Y(NODL)
```


Scanned with OKEN Scanner

```
      DX=XK-XL
      DY=YK-YL
C     LUNGIMEA LATURII CU SCHIMB DE CALDURA PRIN FLUX TERMIC
      DLKL=SQRT(DX*DX+DY*DY)
      DO 3 I=1,3
    3 FEF(I)=0
      H=1.
      IF (NFLAG.NE.1) GOTO 5
      PI=3.1415927
      H=PI*(XK+XL)
    5 CONTINUE
C     POZITIA NODULUI OPUS LATURII
      DO 15 I=1,3
      IF (ID(NREL,I).EQ.NODK) GOTO 15
      IF (ID(NREL,I).EQ.NODL) GOTO 15
      NLC=I
      GOTO 18
   15 CONTINUE
C     VECTORUL PENTRU SCHIMB PRIN FLUX TERMIC
   18 DO 20 I=1,3
      IF (I.EQ.NLC) GOTO 20
      FEF(I)=0.5*H*FLU(NME)*DLKL
   20 CONTINUE
C     ADAUGA CONTRIBUTIA FLUXULUI TERMIC LA VECTORUL FE
      DO 25 I=1,3
      FE(I)=FE(I)+FEF(I)
   25 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ASM (II,JJ,MM)
C     ASAMBLEAZA MATRICELE DE CONDUCTIVITATE SI VECTORII TERMENULUI
C     LIBER ALE ELEMENTELOR IN MATRICEA DE CONDUCTIVITATE SI IN
C     VECTORUL TERMENULUI LIBER AL DOMENIULUI,PE DISC FLEXIBIL
      DIMENSION KK(3),S(3,32)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(400),Y(400),ALAMDX(450),ALAMDY(450),FLINT(450)
      COMMON/BL3/ICON(400),ID(450,3),TETA(400)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(350,3),NFLUX(350,3),ALFA(350),TETAE(350),FLU(350)
      MB1=MBAND+1
      DO 10 I=1,3
      DO 10 J=1,32
   10 S(I,J)=0.
      KK(1)=ICON(II)
      KK(2)=ICON(JJ)
      KK(3)=ICON(MM)
      I=0
   20 I=I+1
      IF (KK(I).EQ.0) GOTO 80
      M=KK(I)
      READ (6,REC=M) (S(I,K),K=1,MB1)
   80 IF (I.LT.NOD) GOTO 20
      DO 150 I=1,NOD
      IF (KK(I).EQ.0) GOTO 150
      M=KK(I)
      S(I,MB1)=S(I,MB1)+FE(I)
      DO 100 J=1,NOD
      IF (KK(J).LT.M) GOTO 100
      NK=KK(J)-M+1
      S(I,NK)=S(I,NK)+SE(I,J)
  100 CONTINUE
  150 CONTINUE
      I=0
  160 I=I+1
      IF (KK(I).EQ.0) GOTO 200
      M=KK(I)
      WRITE (6,REC=M) (S(I,K),K=1,MB1)
  200 IF (I.LT.NOD) GOTO 160
      RETURN
      END
```

```
C     ***********************************
C     * P R O G R A M U L  A T E F S 2 *
C     ***********************************
C     REZOLVAREA PRIN RETROSUBSTITUTIE GAUSS A SISTEMELOR DE ECUATII
C     CU MATRICEA COEFICIENTILOR BANDA DEPUSA PE FLOPPY-DISC IMPREUNA
C                    CU VECTORUL INCARCARILOR
C     (Rezolvarea se face pe blocuri cu dimensiunea MBAND*MBAND.iar
C       in memoria interna se rezerva cimpul SI(2*MBAND,MBAND) si
C                    cimpul RI(2*MBAND))
C
C
C    LISTA NOTATIILOR
C      NSIZE-NUMAR TOTAL DE ECUATII
C      MBAND-SEMILATIMEA BENZII
C      NN-NUMAR DE NODURI
C
C    ALOCAREA MEMORIEI
      DIMENSION ICON(400),TETA(400)
      COMMON/BL1/SI(62,31),RI(62),NSIZE,MBAND,MAC,NRE,NC,NEQ
      COMMON/BL2/SAVE(400)
C    REZOLVAREA SISTEMULUI DE ECUATII AL DOMENIULUI DE ANALIZA
      CALL OPEN (6,'MATRICEAS  ',2)
      REWIND 6
      CALL OPEN (5,'NSIZEMBAND ',2)
      READ (5,REC=1) NSIZE,MBAND,NN
      ENDFILE 5
      DO 5 I=1,62
      RI(I)=0
      DO 5 J=1,31
      SI(I,J)=0
    5 CONTINUE
      NEQ=NSIZE
      NC=NSIZE/MBAND
      ND=NSIZE/MBAND*MBAND-NSIZE
      IF (ND.EQ.0) GOTO 80
      NC=NC+1
   80 CONTINUE
      CALL RED
      CALL SUB
      ENDFILE 6
      DO 100 K=1,NN
      READ (7,1350) KI,X,Y,TETA(K),ICON(K)
  100 CONTINUE
      NSI=0
      DO 110 II=1,NN
      IF (ICON(II).GT.0) GOTO 105
      NSI=NSI+1
      ICON(II)=NSI
      GOTO 110
  105 ICON(II)=0
  110 CONTINUE
      DO 120 I=1,NN
      II=ICON(I)
      IF (II.EQ.0) GOTO 120
      TETA(I)=SAVE(II)
  120 CONTINUE
      WRITE (2,1400)
      DO 130 I=1,NN
      WRITE (2,1500) I,TETA(I)
  130 CONTINUE
      IF (NN.EQ.NN/32*32) GOTO 150
      NS=NN/32+1
      GOTO 155
  150 NS=NN/32
  155 CONTINUE
      CALL OPEN (8,'TEMPERATURI',2)
      I=0
  160 I=I+1
      M=32*I-31
      N=M+31
      WRITE (8,REC=I) (TETA(J),J=M,N)
      IF (I.LE.NS) GOTO 160
      ENDFILE 8
 1350 FORMAT (I5,3F10.2,I5)
 1400 FORMAT (//,6X,'VALOAREA FUNCTIEI NECUNOSCUTE IN NODURI',/,2X,
     *'(TEMPERATURA, GRD. C, PRESIUNEA, M COL. APA)',//,2X,
     *'NR. NOD   VALOAREA FUNCTIEI IN NOD',//)
```


Scanned with OKEN Scanner

```
 1500 FORMAT (1X,I5,5X,E17.7)
      STOP
      END



      SUBROUTINE RED
C     REDUCERE PE BLOCURI
      COMMON/BL1/SI(62,31),RI(62),NSIZE,MBAND,MAC,NRE,NC,NEQ
      COMMON/BL2/SAVE(400)
C     CITESTE PRIMELE DOUA BLOCURI DE PE DISC IN SI,RI (INCARCARE INITIALA)
      MBD=2*MBAND
      LID=1
      LSD=MBD
      LIM=1
      CALL RDI (LID,LSD,LIM)
C     REDUCERE PE BLOCURI
      MAC=1
      NSIZE=MBAND
      DO 180 KK=1,NC
C     APEL SOLVE1 CARE REDUCE BLOCUL CURENT, KK (BL. A)
      CALL SOLVE1
C     TRANSFER 1. BL. CURENT KK DIN A SE SCRIE PE DISC
      LID=(KK-1)*MBAND+1
      LSD=LID+MBAND-1
      LIM=1
      CALL WRI (LID,LSD,LIM)
      IF (KK.NE.1) GOTO 120
      DO 115 I=1,MBAND
  115 SAVE(I)=SI(1,I)
      SAVE(MBAND+1)=RI(1)
  120 CONTINUE
C     TRANSFER 2. BL. B TRECE IN BL. A IN MEMORIA INTERNA
      IF (KK.EQ.NC) GOTO 140
      DO 140 I=1,MBAND
      LI=I+MBAND
      RI(I)=RI(LI)
      DO 140 J=1,MBAND
      SI(I,J)=SI(LI,J)
  140 CONTINUE
C     TRANSFER 3. BL. K+2 DE PE DISC SE CITESTE IN BL. B
      IF (KK.GE.(NC-1)) GOTO 1700
      LID=(KK+1)*MBAND+1
      LSD=LID+MBAND-1
      LIM=MBAND+1
      CALL RDI (LID,LSD,LIM)
  170 CONTINUE
  180 CONTINUE
      RETURN
      END



      SUBROUTINE SUB
C     SUBSTITUTIE GAUSS
      COMMON/BL1/SI(62,31),RI(62),NSIZE,MBAND,MAC,NRE,NC,NEQ
      COMMON/BL2/SAVE(400)
C     SUBSTITUTIA INAPOI BLOC CU BLOC
      MAC=2
      NSIZE=MBAND
      NREST=NC*MBAND-NEQ
C     ULTIMUL BLOC (BL. NC) AL LUI S SI R DE PE DISC SE CITESTE
C     IN BL. A (INCARCARE INITIALA)
      LID=(NC-1)*MBAND+1
      LSD=LID+MBAND-1
      LIM=1
      CALL RDI (LID,LSD,LIM)
      NC1=NC+1
C     SUBSTITUTIA INAPOI
      DO 280 KK=1,NC1
      NRE=1
      IF (KK.NE.1) GOTO 220
      NRE=NREST+2
  220 CONTINUE
      IF (KK.GT.NC) GOTO 225
C     APEL SOLVE1 PENTRU SUBSTITUTIA INAPOI
      CALL SOLVE1
C     TRANSFER 1. RI DIN BL. B SE TRANSFORMA IN VECTORUL R
  225 M=NC-KK+1
      IF (KK-1) 228,235,228
  228 LI=M*MBAND+1
      LIRI=MBAND+1
      DO 230 I=1,MBAND
      SAVE(LI)=RI(LIRI)
      LI=LI+1
      LIRI=LIRI+1
  230 CONTINUE
  235 CONTINUE
C     TRANSER 2. BL. A TRECE IN B,IN MEMORIA INTERNA SE TRANSFERA NUMAI RI.
      IF (KK.GT.NC) GOTO 240
      DO 240 I=1,MBAND
      II=I+MBAND
      RI(II)=RI(I)
  240 CONTINUE
C     TRANSFER 3. BL. KK DE PE DISC SE CITESTE IN BL. A
      IF (KK.GE.NC) GOTO 260
      LID=(M-2)*MBAND+1
      LSD=LID+MBAND-1
      LIM=1
      CALL RDI (LID,LSD,LIM)
  260 CONTINUE
  280 CONTINUE
      RETURN
      END



      SUBROUTINE SOLVE1
      COMMON/BL1/SI(62,31),F(62),NSIZE,MBAND,MAC,NRE,NC,NEQ
C     REDUCEREA S SI F
      GOTO (3,28),MAC
    3 DO 15 N=1,NSIZE
      DO 10 L=2,MBAND
      IF (S(N,L).EQ.0.) GOTO 10
      I=N+L-1
      Z=S(N,L)/S(N,1)
      J=0
      DO 5 K=L,MBAND
      J=J+1
    5 S(I,J)=S(I,J)-Z*S(N,K)
      S(N,L)=Z
   10 CONTINUE
   15 CONTINUE
      DO 25 N=1,NSIZE
      DO 20 L=2,MBAND
      IF (S(N,L).EQ.0.) GOTO 20
      I=N+L-1
      F(I)=F(I)-S(N,L)*F(N)
   20 CONTINUE
   25 F(N)=F(N)/S(N,1)
      RETURN
C     SUBSTITUTIA GAUSS
   28 DO 35 M=NRE,NSIZE
      N=NSIZE+1-M
      DO 30 L=2,MBAND
      IF (S(N,L).EQ.0.) GOTO 30
      K=N+L-1
      F(N)=F(N)-S(N,L)*F(K)
   30 CONTINUE
   35 CONTINUE
      RETURN
      END



      SUBROUTINE RDI (LID,LSD,LIM)
      COMMON/BL1/SI(62,31),RI(62),NSIZE,MBAND,MAC,NRE,NC,NEQ
      I=LID
      J=LIM
    2 READ (6,REC=I) (SI(J,K),K=1,MBAND),RI(J)
      I=I+1
      J=J+1
      IF (I.LE.LSD) GOTO 2
```


Scanned with OKEN Scanner

```
      RETURN
      END


      SUBROUTINE VRI (LID,LSD,LIM)
      COMMON/BL1/SI(62,31),RI(62),NSIZE,MBAND,MAC,NRE,NC,NEQ
      I=LID
      J=LIM
    2 WRITE (6,REC=I) (SI(J,K),K=1,MBAND),RI(J)
      I=I+1
      J=J+1
      IF (I.LE.LSD) GOTO 2
      RETURN
      END


C     ********************************
C     * P R O G R A M U L  A T E F T *
C     ********************************
C     ANALIZA TERMICA CU ELEMENTE FINITE LINIARE IN REGIM TRANZITORIU
C               PENTRU PROBLEME PLANE AXIAL SIMETRICE
C
C
C   ALOCAREA MEMORIEI
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(120),Y(120),ALAMDX(150),ALAMDY(150),FLINT(150)
      COMMON/BL3/ICON(120),ID(150,3),TETA(120),CSP(150),GMA(150)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(110,3),NFLUX(110,3),ALFA(110),TETAE(110),FLU(110)
      COMMON/BL7/S(100,23),F(100)
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      TIMP=0
      NOD=3
      WRITE(7 (3,500)
    2 READ (4,1000) NIT,DTI
      WRITE (3,1100) NIT,DTI
      WRITE (3,1200)
    3 READ (4,1300) NC
      IF (NC.EQ.0) GOTO 2
      IF (NC.EQ.1) GOTO 4
      WRITE (3,1400)
      GOTO 3
    4 CONTINUE
      CALL INE (NFLAG)
      WRITE (2,1500) NIT,DTI
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      DO 15 I=1,NN
      X(I)=X(I)/100
   15 Y(I)=Y(I)/100
      DO 100 JJ=1,NIT
      DO 30 I=1,NSIZE
      F(I)=0
      DO 30 J=1,MBAND
   30 S(I,J)=0
      CALL FMK (NFLAG,DTI)
      TIMP=TIMP+DTI
      WRITE (2,2000) TIMP
      CALL SOL
  100 CONTINUE
  500 FORMAT (2X,'NR. DE INTERVALE DE TIMP SI MARIMEA INTERVALULUI IN ORE',/)
 1000 FORMAT (I5,F7.5)
 1100 FORMAT (2X,I5,F7.5,//)
 1200 FORMAT (2X,'CORECT-1, GRESIT-0',/)
 1300 FORMAT (I5)
 1400 FORMAT (2X,'1 SAU 0',/)
 1500 FORMAT (2X,'NR. DE INTERVALE DE TIMP',I3,/,2X,
     *'MARIMEA INTERVALULUI DE TIMP',F7.5,' ORE')
 2000 FORMAT (///,2X,'TEMPERATURILE LA MOMENTUL ',F7.5,' ORE')
      STOP
      END


      SUBROUTINE TIM (NFLAG,ME,II,JJ,MM,DTI)
      DIMENSION CS(3,3),SES(3,3),TETAV(3)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(120),Y(120),ALAMDX(150),ALAMDY(150),FLINT(150)
      COMMON/BL3/ICON(120),ID(150,3),TETA(120),CSP(150),GMA(150)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(110,3),NFLUX(110,3),ALFA(110),TETAE(110),FLU(110)
      COMMON/BL7/S(100,23),F(100)
      IF (NFLAG.NE.1) GOTO 10
      PI=3.1415927
      D=2.*PI*DELTA*GMA(ME)*CSP(ME)/180
      RI=X(II)
      RJ=X(JJ)
      RK=X(MM)
      CS(1,1)=D*(12.*RI*RI+2.*RJ*RJ+2.*RK*RK+6.*RI*RJ+6.*RI*RK+2.*RJ*RK)
      CS(1,2)=D*(3.*RI*RI+3.*RJ*RJ+RK*RK+4.*RI*RJ+2.*RI*RK+2.*RJ*RK)
      CS(1,3)=D*(3.*RI*RI+RJ*RJ+3.*RK*RK+2.*RI*RJ+6.*RI*RK+2.*RJ*RK)
      CS(2,2)=D*(2.*RI*RI+12.*RJ*RJ+2.*RK*RK+6.*RI*RJ+2.*RI*RK+6.*RJ*RK)
      CS(2,3)=D*(RI*RI+3.*RJ*RJ+3.*RK*RK+2.*RI*RJ+2.*RI*RK+4.*RJ*RK)
      CS(3,3)=D*(2.*RI*RI+2.*RJ*RJ+12.*RK*RK+2.*RI*RJ+6.*RI*RK+6.*RJ*RK)
      CS(2,1)=CS(1,2)
      CS(3,1)=CS(1,3)
      CS(3,2)=CS(2,3)
      GOTO 18
   10 D=GMA(ME)*CSP(ME)*DELTA/12.
      DO 15 I=1,3
      DO 15 J=1,3
      IF (I.EQ.J) GOTO 13
      CS(I,J)=D
      GOTO 15
   13 CS(I,J)=2.*D
   15 CONTINUE
   18 CONTINUE
      DO 20 I=1,3
      DO 20 J=1,3
   20 SES(I,J)=SE(I,J)
      DO 25 I=1,3
      DO 25 J=1,3
   25 SE(I,J)=CS(I,J)+2.*DTI/3.*SE(I,J)
      TETAV(1)=TETA(II)
      TETAV(2)=TETA(JJ)
      TETAV(3)=TETA(MM)
      DO 30 I=1,3
      DO 30 J=1,3
   30 SES(I,J)=CS(I,J)-DTI/3.*SES(I,J)
      DO 40 I=1,3
      SUM=0.
      DO 35 J=1,3
   35 SUM=SUM+SES(I,J)*TETAV(J)
      FE(I)=SUM+FE(I)*DTI
   40 CONTINUE
      RETURN
      END


      SUBROUTINE ASM (II,JJ,MM)
      DIMENSION KK(3)
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(120),Y(120),ALAMDX(150),ALAMDY(150),FLINT(150)
      COMMON/BL3/ICON(120),ID(150,3),TETA(120),CSP(150),GMA(150)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(110,3),NFLUX(110,3),ALFA(110),TETAE(110),FLU(110)
      COMMON/BL7/S(100,23),F(100)
      KK(1)=ICON(II)
      KK(2)=ICON(JJ)
      KK(3)=ICON(MM)
      DO 15 I=1,NOD
      IF (KK(I).EQ.0) GOTO 15
      LL=KK(I)
      F(LL)=F(LL)+FE(I)
      DO 10 J=1,NOD
      IF (KK(J).LT.LL) GOTO 10
      NK=KK(J)-LL+1
      S(LL,NK)=S(LL,NK)+SE(I,J)
   10 CONTINUE
```


Scanned with OKEN Scanner

```
   15 CONTINUE
      RETURN
      END


      SUBROUTINE SOL
      COMMON/BL1/NN,NE,NOD,NPROBL,NSIZE,MBAND,NNTIMP,NECONV,NEFLUX
      COMMON/BL2/X(120),Y(120),ALAMDX(150),ALAMDY(150),FLINT(150)
      COMMON/BL3/ICON(120),ID(150,3),TETA(120),CSP(150),GMA(150)
      COMMON/BL4/SE(3,3),FE(3)
      COMMON/BL5/A(3),B(3),C(3),DELTA,DET
      COMMON/BL6/NCON(110,3),NFLUX(110,3),ALFA(110),TETAE(110),FLU(110)
      COMMON/BL7/S(100,23),F(100)
      DO 15 N=1,NSIZE
      DO 10 L=2,MBAND
      IF (S(N,L).EQ.0.) GOTO 10
      I=N+L-1
      Z=S(N,L)/S(N,1)
      J=0
      DO 5 K=L,MBAND
      J=J+1
    5 S(I,J)=S(I,J)-Z*S(N,K)
      S(N,L)=Z
      F(I)=F(I)-Z*F(N)
   10 CONTINUE
      F(N)=F(N)/S(N,1)
   15 CONTINUE
      DO 35 M=2,NSIZE
      N=NSIZE+1-M
      DO 30 L=2,MBAND
      IF (S(N,L).EQ.0.) GOTO 30
      K=N+L-1
      F(N)=F(N)-S(N,L)*F(K)
   30 CONTINUE
   35 CONTINUE
      DO 40 I=1,NN
      II=ICON(I)
      IF (II.EQ.0) GOTO 40
      TETA(I)=F(II)
   40 CONTINUE
      WRITE (2,100)
      DO 50 I=1,NN
   50 WRITE (2,200) I,TETA(I)
  100 FORMAT (////,2X,'NN   TEMPERATURA, GRD. C',//)
  200 FORMAT (1X,I3,10X,E15.7)
      RETURN
      END



C     ********************************
C     *PROGRAMUL FMIN1*
C     ********************************
C     DETERMINAREA MINIMULUI UNEI FUNCTII MONOVARIABILE DUPA ALGORITMUL
C        DE CAUTARE CU PAS ACCELERAT CUPLAT CU INTERPOLARE PATRATICA
C
C
C   LISTA NOTATIILOR
C     X(1)-PUNCTUL DE START
C     P-PASUL
C     R-COEFICIENTUL DE REDUCERE A PASULUI IN CAZUL INTERPOLARII NEREUSITE
C     EPS-VALOAREA CRITICA DE CONVERGENTA
C     XO-REZULTATUL INTERPOLARII
C
C   ALOCAREA MEMORIEI
      DIMENSION X(4),F(4)
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) X(1),P,R,EPS
    1 FORMAT (3F5.2,F10.6)
      WRITE (3,2)
    2 FORMAT (////,25X,30('*')./,25X,'*PROGRAMUL FMIN1*',
     */,25X,30('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,3) X(1),P,R,EPS
    3 FORMAT (/,19X,'X(1)=',F5.2,2X,'P=',F5.2,2X,'R=',F5.2,2X,'EPS=',F8.6)
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,4)
    4 FORMAT (//.29X.'REZULTATELE CALCULULUI',/,29X,22('*'))
C     CALCULAREA PRIN PASI
    5 I=1
      J=0
      K=0
      CALL FUNCTIA (I,X,F)
    6 X(2)=X(1)+P
      I=2
      CALL FUNCTIA (I,X,F)
      IF (F(2).GE.F(1)) GOTO 8
    7 X(3)=X(2)+2*(X(2)-X(1))
      I=3
      CALL FUNCTIA (I,X,F)
      IF (F(3).GE.F(2)) GOTO 11
      X(1)=X(2)
      X(2)=X(3)
      F(2)=F(3)
      GOTO 7
    8 IF (F(2).EQ.F(1)) GOTO 9
      IF (K.EQ.1) GOTO 10
      P=-P
      K=1
      I=1
      GOTO 6
    9 X(3)=(X(1)+X(2))/2
      I=2
      CALL FUNCTIA (I,X,F)
      GOTO 11
   10 X(3)=X(1)-(X(1)-X(2))
      I=2
      CALL FUNCTIA (I,X,F)
C     IDENTIFICAREA CELUI MAI NEFAVORABIL PUNCT
   11 KB=2
      DO 12 I=1,3
      IF (F(KB).LE.F(I)) GOTO 15
   12 CONTINUE
      X(4)=(X(2)**2-X(3)**2)*F(1)+(X(3)**2-X(1)**2)*F(2)
      X(4)=X(4)+(X(1)**2-X(2)**2)*F(3)
      W=(X(2)-X(3))*F(1)+(X(3)-X(1))*F(2)+(X(1)-X(2))*F(3)
      IF (W.EQ.0) GOTO 13
      X(4)=X(4)/(2*W)
      I=4
      CALL FUNCTIA (I,X,F)
      IF (J.EQ.0) GOTO 14
C     VERIFICAREA REUSITEI INTERPOLARII
      IF (F(4).LT.F0) GOTO 14
   13 X(1)=XS
      P=P/R
      GOTO 5
C     VERIFICAREA CONVERGENTEI
   14 IF (ABS(F(4)-F0).LE.EPS) GOTO 16
      X(KB)=X(4)
      F(KB)=F(4)
      F0=F(4)
      XS=X(4)
      J=1
      GOTO 11
   15 KB=I
      GOTO 12
   16 WRITE (3,17) X(4),F(4)
   17 FORMAT (//,27X,'XOPT=',F8.5,/,27X,'FMIN=',F12.6)
      STOP
      END
      SUBROUTINE FUNCTIA (I,X,F)
C     FUNCTIA DE OPTIMIZAT
      DIMENSION X(4),F(4)
      F(1)=-1-8*X(1)-2*X(1)**2+10*X(1)**3/3+
     *X(1)**4/4-4*X(1)**5/5+X(1)**6/6
      RETURN
      END
```


Scanned with OKEN Scanner

```
C     ********************************
C     * P R O G R A M U L  F M I N 2 *
C     ********************************
C     DETERMINAREA MINIMULUI UNEI FUNCTII MONOVARIABILE DUPA
C                    ALGORITMUL BRENT
C
C
C    LISTA NOTATIILOR
C      XMIN,XMAX-LIMITELE INTERVALULUI IN CARE SE CAUTA MINIMUL
C      T,EPS-PRECIZIA RELATIVA SI EROAREA ABSOLUTA CU CARE SE CALCULEAZA
C           MINIMUL FUNCTIEI F(X)
C      LIM-NUMARUL LIMITA ADMIS PENTRU APELUL FUNCTIEI F(X)
C      X,FX-VALOAREA VARIABILEI SI A FUNCTIEI PENTRU PUNCTUL DE MINIM
C      KOD-IDENTIFICATOR CU VALOAREA 0 SAU 1 DUPA CUM S-A OBTINUT SAU NU
C         MINIMUL INTR-UN NUMAR DE APELURI ALE FUNCTIEI F(X) MAI MIC
C         SAU EGAL CU LIM
C
C    FUNCTIA DE MINIMIZAT
      F(X)=0.005*X+100.*(20.1+2*X)/(20.1+400.*X)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) XMIN,XMAX,T,EPS,LIM
    1 FORMAT (4F10.6,I5)
      WRITE (3,2)
    2 FORMAT (////,25X,30('*'),/,25X,'*P R O G R A M U L  F M I N 2*',
     */,25X,30('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,3) XMIN,XMAX,T,EPS,LIM
    3 FORMAT (/,2X,'XMIN=',E14.6,5X,'XMAX=',E14.6,5X,'T=',E14.6,
     */,17X,'EPS=',E14.6,5X,'LIM=',I4)
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,4)
    4 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      KOD=0
      NEF=0
      X=XMIN+((3*SQRT(5))/2)*(XMAX-XMIN)
      W=X
      V=X
      E=0.
      FX=F(X)
      NEF=NEF+1
      FW=FX
      FV=FX
    5 C=0.5*(XMIN+XMAX)
      ER=T*ABS(X)+EPS
      IF (ABS(X-C).LE.2*ER-0.5*(XMAX-XMIN)) GOTO 24
      R=0.
      Q=0.
      P=0.
      IF (ABS(E).LE.ER) GOTO 8
      R=(X-W)*(FX-FV)
      Q=(X-V)*(FX-FW)
      P=(X-V)*Q-(X-W)*R
      Q=2*(Q-R)
      IF (Q.LE.0.) GOTO 6
      P=-P
      GOTO 7
    6 Q=-Q
    7 R=E
      E=D
    8 IF (ABS(P).GE.ABS(0.5*Q*R)) GOTO 10
      IF (P.LE.Q*(XMIN-X).OR.(P.GE.Q*(XMAX-X))) GOTO 10
      D=P/Q
      U=X+D
      IF ((U-XMIN.GE.2*ER).AND.(XMAX-U.GE.2*ER)) GOTO 13
      IF (X.GE.C) GOTO 9
      D=ER
      GOTO 13
    9 D=-ER
      GOTO 13
   10 IF (X.GE.C) GOTO 11
      E=XMAX-X
      GOTO 12
   11 E=XMIN-X
   12 D=((3.-SQRT(5.))/2.)*E
   13 IF (ABS(D).LT.ER) GOTO 14
      U=X+D
      GOTO 16
   14 IF (D.LE.0.) GOTO 15
      U=X+ER
      GOTO 16
   15 U=X-ER
   16 FU=F(U)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 23
      IF (FU.GT.FX) GOTO 19
      IF (U.GE.X) GOTO 17
      XMAX=X
      GOTO 18
   17 XMIN=X
   18 V=W
      FV=FW
      W=X
      FW=FX
      X=U
      FX=FU
      GOTO 5
   19 IF (U.GE.X) GOTO 20
      XMIN=U
      GOTO 21
   20 XMAX=U
   21 IF ((FU.GT.FV).AND.(W.NE.X)) GOTO 22
      V=W
      FV=FW
      W=U
      FW=FU
      GOTO 5
   22 IF ((FU.GT.FV).AND.(V.NE.X).AND.(V.NE.W)) GOTO 5
      V=U
      FV=FU
      GOTO 5
   23 KOD=1
   24 WRITE (3,25) KOD,X,FX
   25 FORMAT (//,6X,'KOD=',I1,5X,'X=',E16.8,5X,'FMIN=',E16.8)
      STOP
      END
```

```
C     ********************************
C     * P R O G R A M U L  F M I N 3 *
C     ********************************
C     DETERMINAREA MINIMULUI UNEI FUNCTII DE N VARIABILE CU METODA
C             GRADIENTILOR CONJUGATI (FLETCHER-REEVES)
C
C
C    LISTA NOTATIILOR
C      N-NUMARUL VARIABILELOR INDEPENDENTE
C      EST-VALOAREA ESTIMATIVA A MINIMULUI FUNCTIEI F(X)
C      EPS-PRECIZIA CU CARE SE CALCULEAZA MINIMUL
C      LIM-NUMARUL LIMITA ADMIS AL APELURILOR SUBRUTINEI FUNCT
C      NIT-NUMARUL ITERATIILOR PARCURSE
C      X(I)-VECTORUL CONTININD INITIAL PUNCTUL DE START, IAR IN FINAL
C           VALORILE VARIABILELOR PENTRU CARE S-A OBTINUT MINIMUL
C      KOD-IDENTIFICATOR CU VALORILE :0 SAU 1-DUPA CUM S-A OBTINUT SAU
C          NU MINIMUL INTR-UN NUMAR DE APELURI ALE FUNCTIEI MAI MIC
C          SAU EGAL CU LMIN; 2-CIND NU EXISTA MINIM; 3-CIND EXISTA
C          ERORI IN CALCULUL COMPONENTELOR GRADIENTULUI
C
C    ALOCAREA MEMORIEI
      DIMENSION X(250),G(250),D(500)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,EST,EPS,LIM,(X(I),I=1,N)
    1 FORMAT (I5,F10.5,F15.12,I5,/,(6F10.5))
      WRITE (3,2)
    2 FORMAT (////,25X,30('*'),/,25X,'*P R O G R A M U L  F M I N 3*',
     */,25X,30('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,3) N,EST,EPS,LIM
    3 FORMAT (/,2X,'N=',I3,5X,'EST=',E16.8,5X,'EPS=',E16.8,'LIM=',I4)
      WRITE (3,4) (X(I),I=1,N)
    4 FORMAT (/,2X,'VALORILE INITIALE ALE VARIABILELOR',//,2X,(4E16.3))
```


Scanned with OKEN Scanner

```
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,5)
    5 FORMAT (///,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      NEF=0
      CALL FUNCT (N,X,F,G)
      NEF=NEF+1
      NIT=0
      KOD=0
      N1=N+1
    6 DO 32 L=1,N1
      NIT=NIT+1
      FY=F
      GP=0.
      DO 7 J=1,N
    7 GP=GP+G(J)*G(J)
      IF (GP.EQ.0) GOTO 35
      IF (L.NE.1) GOTO 9
      DO 8 J=1,N
    8 D(J)=-G(J)
      GOTO 11
    9 BETA=GP/GPV
      DO 10 J=1,N
   10 D(J)=BETA*D(J)-G(J)
   11 DY=0.
      SN=0.
      DO 12 J=1,N
      X=J+N
      D(X)=X(J)
      SN=SN+ABS(D(J))
   12 DY=DY+D(J)*G(J)
      IF (DY.GE.0.) GOTO 31
      SNR=1./SN
      FY=F
      PAS=2.*(EST-F)/DY
      R=SNR
      IF (PAS.LE.0.) GOTO 13
      IF (PAS.GE.R) GOTO 13
      R=PAS
   13 PAS=0.
   14 FX=FY
      DX=DY
      DO 15 I=1,N
   15 X(I)=X(I)+R*D(I)
      CALL FUNCT (N,X,F,G)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 33
      FY=F
      DY=0.
      DO 16 I=1,N
   16 DY=DY+G(I)*D(I)
      IF (DY) 17,28,19
   17 IF (FY.GE.FX) GOTO 19
      R=R+PAS
      PAS=R
      IF (SN*R.LE.1*E20) GOTO 14
   18 KOD=2
      GOTO 39
   19 T=0.
   20 IF (R.EQ.0.) GOTO 28
      Z=3.*(FX-FY)/R+DX+DY
      IF (Z**2.LT.DX*DY) GOTO 36
      Q=SQRT(Z**2-DX*DY)
      PAS=(DY+Q-Z)*R/(DX-DY+2*Q)
      DO 21 I=1,N
   21 X(I)=X(I)+(T-PAS)*D(I)
      CALL FUNCT (N,X,F,G)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 33
      IF (F.GT.FX) GOTO 22
      IF (F.LE.FY) GOTO 28
   22 DPAS=0.
      DO 23 I=1,N
   23 DPAS=DPAS+G(I)*D(I)
      IF (DPAS.GE.0.) GOTO 26
      IF (F-FX) 25,24,26
   24 IF (DX.EQ.DPAS) GOTO 28
   25 FX=F
      DX=DPAS
      T=PAS
      R=PAS
      GOTO 20
   26 IF (FY.NE.F) GOTO 27
      IF (DY.EQ.DPAS) GOTO 28
   27 FY=F
      DY=DPAS
      R=R-PAS
      GOTO 20
   28 T=0.
      DO 29 J=1,N
      K=J+N
      D(K)=X(J)-D(K)
   29 T=T+ABS(D(K))
      IF (NIT.LT.N1) GOTO 30
      IF (T.LE.EPS) GOTO 34
   30 IF (FY-F+EPS) 18,38,31
   31 GPV=GP
   32 KOD=0
      GOTO 6
   33 KOD=1
      GOTO 39
   34 IF (GP.LT.EPS) GOTO 38
   35 KOD=0
      GOTO 39
   36 DO 37 J=1,N
      K=N+J
   37 X(J)=D(K)
      CALL FUNCT (N,X,F,G)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 33
   38 IF (KOD.NE.0) GOTO 39
      KOD=3
      GOTO 6
   39 WRITE (3,40) KOD,NIT,F,(X(I),I=1,N)
   40 FORMAT (//,29X,'KOD=',I1,5X,'ITER=',I4,//,30X,'FMIN=',E16.8,
     *//,23X,'VALORILE VARIABILELOR PENTRU FMIN:',/,4(2X,E16.8))
      STOP
      END


      SUBROUTINE FUNCT (N,X,F,G)
      DIMENSION X(250),G(250)
C     FUNCTIA DE MINIMIZAT
      F=100.*(X(2)-X(1)**2)**2+(1.-X(1))**2
C     RELATIILE DE CALCUL ALE COMPONENTELOR GRADIENTULUI
      G(1)=-400.*X(1)*X(2)+400.*X(1)**3-2.+2.*X(1)
      G(2)=200.*X(2)-200.*X(1)**2
      RETURN
      END


C     ***********************************
C     *  P R O G R A M U L   S I M P L E X *
C     ***********************************
C     DETERMINAREA OPTIMULUI FUNCTIEI OBIECTIV IN PROBLEME DE
C        PROGRAMARE LINIARA CU ALGORITMUL SIMPLEX PRIMAL
C
C
C   LISTA NOTATIILOR
C     N-NUMARUL VARIABILELOR INDEPENDENTE
C     M1-NUMARUL RESTRICTIILOR INEGALITATI DE FORMA:B+SUM(AX).GE.0
C     M2-NUMARUL RESTRICTIILOR INEGALITATI DE FORMA:B+SUM(AX).LE.0
C     M3-NUMARUL RESTRICTIILOR EGALITATI DE FORMA:B+SUM(AX).EQ.0
C     M-NUMARUL TOTAL AL RESTRICTIILOR (M=M1+M2+M3)
C     PL,PC-PASUL DE LINIE SI DE COLOANA CU VALORILE:
C           PL=N+1;PC=1-LA CITIREA COEFICIENTILOR PE LINIE:
C           PL=1;PC=N+1-LA CITIREA COEFICIENTILOR PE COLOANE
C     IM-IDENTIFICATOR CU VALORILE 0 SAU 1,DUPA CUM SE CAUTA
C        MAXIMUL SAU MINIMUL FUNCTIEI OBIECTIV
C     W(I)-TABLOUL SIMPLEX,CUPRINZIND COEFICIENTII FUNCTIEI OBIECTIV
C          PE PRIMA LINIE,IAR APOI PE CEI AI RESTRICTIILOR M1,M2,M3,
```


Scanned with OKEN Scanner

```
C        SCRISE ASTFEL CA TERMENII LIBERI SA APARA PE PRIMA COLOANA.
C        PLUS COEFICIENTI  0  PE O ULTIMA LINIE,PENTRU FUNCTIA
C        OBIECTIV ARTIFICIALA
C     K-DIMENSIUNEA TABLOULUI W(I),AVIND VALOAREA (M+2)*(N+1)
C     EPS-EROAREA ADMISA,CU VALOAREA MINIMA SUM(W(I)/K*10**(-5)
C     IN-IDENTIFICATOR EGAL CU 0 DACA FUNCTIA OBIECTIV SE AFLA PE PRIMA
C       LINIE (CAZUL FUNCTIEI OBIECTIV DATE) SAU EGAL CU M+1 DACA
C       FUNCTIA OBIECTIV SE AFLA PE ULTIMA LINIE (CAZUL FUNCTIEI
C       OBIECTIV ARTIFICIALE)
C     KOD-IDENTIFICATOR CU VALOAREA 0,1 SAU 2,DUPA CUM SOLUTIA PROBLE-
C        MEI ESTE CORECTA,NU EXISTA SOLUTIE FINITA,RESPECTIV NU EXISTA
C        O SOLUTIE ADMISIBILA DE BAZA
C
C     ALOCAREA MEMORIEI
      DIMENSION W(3000),L1(30),L2(100),L3(100),K1(30),K2(100)
      INTEGER PL,PC,R,S,T
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,M1,M2,M3,PL,PC,EPS,IN
    1 FORMAT (6I5,F10.7,I5)
      READ (1,2) K,(W(I),I=1,K)
    2 FORMAT (I5,/,(6F10.6))
      WRITE (3,3)
    3 FORMAT (1///,23X,34('*'),/,23X,'*P R O G R A M U L  S I M P L E X*',
     *,/,23X,34('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,4) N,M1,M2,M3,PL,PC,EPS,IN
    4 FORMAT (/,24X,'N=',I3,3X,'M1=',I3,3X,'M2=',I3,3X,'M3=',I3,/,
     *24X,'PL=',I3,3X,'PC=',I3,3X,'EPS=',F9.7,3X,'IN=',I1)
      WRITE (3,5) (W(I),I=1,K)
    5 FORMAT (//,15X,'TABLOUL SIMPLEX INITIAL',/,(5(3X,F10.6)))
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,6)
    6 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'))
      M=M1+M2+M3
      IF (IN.EQ.0.) GOTO 8
      N1=N+1
      DO 7 I=2,N1
    7 W(I)=-W(I)
    8 R=0
      V=-1
      DO 9 I=1,N
      L1(I)=I
    9 K1(I)=I
      L10=N
      DO 10 I=1,M
      L2(I)=I
   10 K2(I)=N+I
      L20=M
      IF (M2+M3.EQ.0.) GOTO 27
      DO 11 I=1,M
   11 L3(I)=0
      IF (M2.EQ.0.) GOTO 13
      DO 12 I=1,M2
   12 L3(I)=1
   13 R=1
      N1=N+1
      DO 15 I=1,N1
      I1=I-1
      Q1=0.
      M11=M1+1
      DO 14 J=M11,M
      J1=J*PL+I1*PC+1
   14 Q1=Q1+W(J1)
      J2=(M+1)*PL+I1*PC+1
   15 W(J2)=-Q1
   16 CALL COLPIV (W,K,L1,N,L10,M+1,PL,PC,T,WMAX)
      IF (ABS(WMAX).LT.EPS) WMAX=0
      J=(M+1)*PL+1
      IF (ABS(W(J)).LT.EPS) W(J)=0
      IF (WMAX.GT.0.OR.W(J).GE.0.) GOTO 17
      KOD=2
      GOTO 29
   17 IF (WMAX.GT.0.OR.W(J).NE.0.) GOTO 18
      R=0
      GOTO 27
   18 CALL LINPIV (W,K,L2,M,L20,PL,PC,N,S,T,P,V)
      IF (S.NE.0.) GOTO 19
      KOD=2
      GOTO 30
   19 CALL PIVOT (W,K,0,M+1,0,N,PL,PC,S,T)
      IF (K2(S).LT.N+M1+M2+1) GOTO 23
      DO 20 I=1,L10
      IF (L1(I).EQ.T) GOTO 21
   20 CONTINUE
   21 L10=L10+1
      DO 22 J=I,L10
   22 L1(J)=L1(J+1)
      GOTO 24
   23 IF (K2(S).LT.N+M1+1) GOTO 26
      J=K2(S)-M1-N
      IF (L3(J).EQ.0) GOTO 26
      L3(J)=0
   24 J=(M+1)*PL+T*PC+1
      W(J)=W(J)+1
      J1=M+2
      DO 25 I=1,J1
      J=(I-1)*PL+T*PC+1
   25 W(J)=-W(J)
   26 L=K1(T)
      K1(T)=K2(S)
      K2(S)=L
      IF (R.NE.0.) GOTO 16
   27 CALL COLPIV (W,K,L1,N,L10,0,PL,PC,T,WMAX)
      IF (ABS(WMAX).LT.EPS) WMAX=0
      IF (WMAX.GT.0.) GOTO 28
      KOD=0
      GOTO 30
   28 CALL LINPIV (W,K,L2,M,L20,PL,PC,N,S,T,P,V)
      IF (S.NE.0.) GOTO 29
      KOD=1
      GOTO 30
   29 CALL PIVOT (W,K,0,M,0,N,PL,PC,S,T)
      GOTO 26
   30 WRITE (3,31) KOD
   31 FORMAT (//,20X,'CODUL DE SEMNALARE A ERORII KOD=',I2)
      IF (KOD.NE.0.) GOTO 33
      IF (IN.NE.0.) W(1)=-W(1)
      WRITE (3,32) W(1),(K2(I),W(PL-I+1),I=1,M),(K1(I),I=1,N)
   32 FORMAT (//,7X,'VALOAREA OPTIMA A FUNCTIEI OBIECTIV FOPT=',E14
     *//,29X,'VALORILE VARIABILELOR:',/,2X,4(2X,'X(',I2,')=',F8.4),
     */,2X,3(2X,'X(',I2,')=  0.',4X))
   33 STOP
      END

      SUBROUTINE COLPIV (W,K,L1,JL1,L10,IN,PL,PC,T,WMAX)
C     DETERMINAREA COLOANEI PIVOT PENTRU FIECARE ITERATIE
      DIMENSION W(K),L1(JL1)
      INTEGER PL,PC,T
      J=IN*PL+L1(1)*PC+1
      WMAX=W(J)
      T=L1(1)
      IF (L10.LT.2) RETURN
      DO 1 I=2,L10
      J=IN*PL+L1(I)*PC+1
      IF (W(J).LE.WMAX) GOTO 1
      WMAX=W(J)
      T=L1(I)
    1 CONTINUE
      RETURN
      END

      SUBROUTINE LINPIV (W,K,L2,JL2,L20,PL,PC,N,S,T,P,V)
C     DETERMINAREA LINIEI PIVOT PENTRU FIECARE ITERATIE
      DIMENSION W(K),L2(JL2)
      INTEGER PL,PC,R,S,T
      S=0
      IF (L20.LT.1) RETURN
      DO 1 J=1,L20
      I=L2(J)*PL+1
      I1=I+I*PC
```


Scanned with OKEN Scanner

```
      IF (V*W(I1).GT.0.) GOTO 2
    1 CONTINUE
      RETURN
    2 P=V*W(I)/W(I1)
      S=L2(J)
      J1=J+1
      IF (J1.GT.L20) RETURN
      DO 6 J=J1,L20
      I=L2(J)*PL+1
      I1=I+I*PC
      IF (V*W(I1).LE.0.) GOTO 6
      P1=V*W(I)/W(I1)
      IF (P1.GE.P) GOTO 3
      S=L2(J)
      P=P1
      GOTO 6
    3 IF (P1.NE.P) GOTO 6
      I0=L2(J)
      DO 4 R=1,N
      I1=S*PL+R*PC+1
      I11=S*PL+T*PC+1
      I12=I0*PL+R*PC+1
      Q=V*W(I1)/W(I11)
      Q1=V*W(I12)/W(I1)
      IF (Q.LT.Q1) GOTO 6
      IF (Q1.LT.Q) GOTO 5
    4 CONTINUE
    5 S=I0
    6 CONTINUE
      RETURN
      END

      SUBROUTINE PIVOT (W,K,I0,I1,J0,J1,PL,PC,S,T)
C     EFECTUAREA OPERATIEI DE PIVOTARE PENTRU FIECARE ITERATIE
      DIMENSION W(K)
      INTEGER PL,PC,R,S,T
      L=S*PL+T*PC+1
      PIV=1./W(L)
      II0=I0+1
      III=I1+1
      JJ0=J0+1
      JJ1=J1+1
      DO 2 I=II0,III
      II=I-1
      IF (II.EQ.S) GOTO 2
      IA=II*PL+T*PC+1
      W(IA)=W(IA)*PIV
      DO 1 J=JJ0,JJ1
      JJ=J-1
      IF (JJ.EQ.T) GOTO 1
      IB=II*PL+JJ*PC+1
      IC=S*PL+JJ*PC+1
      W(IB)=W(IB)-W(IC)*W(IA)
    1 CONTINUE
    2 CONTINUE
      DO 3 J=JJ0,JJ1
      JJ=J-1
      R=S*PL+JJ*PC+1
      IF (JJ.NE.1) W(R)=-W(R)*PIV
    3 CONTINUE
      A(L)=PIV
      RETURN
      END



C     ************************************
C     * P R O G R A M U L  A R B M I N *
C     ************************************
C     DETERMINAREA ARBORELUI PARTIAL DE VALOARE MINIMA INTR-UN GRAF
C                     CU ALGORITMUL KRUSKAL
C
C
C   LISTA NOTATIILOR
C     NA(I),NB(I)-VECTORI ALE CAROR COMPONENTE REPREZINTA VIRFURILE
C                 MUCHIILOR I DIN GRAF
C     LAM(I)-VECTORUL VALORILOR MUCHIILOR I ALE GRAFULUI
C     MARC(I)-VECTOR ALE CARUI COMPONENTE DAU MARCILE MUCHIILOR
C             GRAFULUI
C     NAA(I),NBA(I)-VECTORI ALE CAROR COMPONENTE REPREZINTA VIRFURILE
C                   MUCHIILOR I ALE ARBORELUI MINIM
C     LAMA(I)-VECTORUL VALORILOR MUCHIILOR I ALE ARBORELUI MINIM
C     NAM(I),NBM(I),LAMM(I)-TABLOURI FOLOSITE LA INTRODUCEREA DATELOR
C
C   ALOCAREA MEMORIEI
      DIMENSION NA(600),NB(600),MARC(600),NAM(4),NBM(4),NAA(250),NBA(250)
      REAL LAM(600),LAMM(4),LAMA(250)
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      WRITE (3,1)
    1 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  A R B M I N*'.
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'),
     */,3X,64('-'),//,2X,4(2X,'NA',2X,'NB',3X,'LAM',2X),//,3X,64('-'))
      IX=0
      IT=0
   99 READ (1,2,END=8) ((NAM(I),NBM(I),LAMM(I)),I=1,4)
    2 FORMAT (4(2I5,F6.1))
      DO 6 I=1,4
      IF (LAMM(1).EQ.0.) GOTO 6
      IF (NAM(I).NE.NBM(I)) GOTO 5
    3 WRITE (3,4) ((NAM(J),NBM(J),LAMM(J)),J=1,4)
    4 FORMAT (/,33X,'DATE ERONATE:',/,3X,4(2I4,F6.1,2X))
      GOTO 24
    5 IF (NAM(I).LT.0.OR.NBM(I).LT.0.OR.LAMM(I).LT.0.) GOTO 3
      IX=IX+1
      NA(IX)=NAM(I)
      NB(IX)=NBM(I)
      LAM(IX)=LAMM(I)
      IX=IX+1
      NA(IX)=NBM(I)
      NB(IX)=NAM(I)
      LAM(IX)=LAMM(I)
    6 CONTINUE
      WRITE (3,7) ((NAM(I),NBM(I),LAMM(I)),I=1,4)
    7 FORMAT (3X,4(2I4,F6.1,2X))
      GOTO 99
    8 WRITE (3,9)
    9 FORMAT (2X,64('-'))
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,10)
   10 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'),/33X,
     *'ARBORELE MINIM',/,33X,14('-'),//,10X,'ITERATIA',6X,'ARCUL',6X,
     *'VALOAREA',/)
      DO 11 I=1,IX
   11 MARC(I)=0
      JAL=0
   12 DO 13 J=1,IX
   13 IF (NB(J).EQ.JAL) MARC(J)=1
      IAL=JAL
      JAL=0
      MIN=9999
      DO 14 I=1,IX
      IF (NA(I).NE.IAL) GOTO 14
      IF (MARC(I).EQ.1) GOTO 14
      IF (LAM(I).GE.MIN) GOTO 14
      JAL=NB(I)
      MIN=LAM(I)
      II=I
   14 CONTINUE
      IF (JAL.EQ.0) GOTO 17
   15 IT=IT+1
      NAA(IT)=NA(II)
      NBA(IT)=NB(II)
      LAMA(IT)=MIN
      IAL=JAL
      WRITE (3,16) IT,NAA(IT),NBA(IT),LAMA(IT)
   16 FORMAT (12X,I4,8X,I3,'-',I3,6X,F8.2)
      GOTO 12
   17 SWEX=0
      DO 18 I=1,IX
   18 IF (MARC(I).EQ.0) SWEX=1
      IF (SWEX.EQ.0) GOTO 21
```


Scanned with OKEN Scanner

```
      IPR=IT+1
      NAA(IPR)=NBA(IT)
      DO 19 I=1,IT
      II=I+1
      DO 19 J=II,IPR
      DO 19 K=1,IX
      IF (LAM(K).EQ.9999) GOTO 19
      IF (NA(K).EQ.NAA(I).AND.NB(K).EQ.NAA(J)) LAM(K)=9999
      IF (NA(K).EQ.NAA(J).AND.NB(K).EQ.NAA(I)) LAM(K)=9999
   19 CONTINUE
      MIN=9999
      DO 20 I=1,IT
      JJ=NAA(I)
      DO 20 J=1,IX
      IF (LAM(J).GE.9999) GOTO 20
      IF (MARC(J).EQ.1) GOTO 20
      IF (NA(J).NE.JJ) GOTO 20
      IF (LAM(J).EQ.9999) GOTO 20
      IF (LAM(K).EQ.9999) GOTO 19
      MIN=LAM(J)
      JAL=NB(J)
      II=J
   20 CONTINUE
      GOTO 15
   21 VAL=0
      DO 22 I=1,IT
   22 VAL=VAL+LAMA(I)
      WRITE (3,23) VAL
   23 FORMAT (//,12X,'VALOAREA TOTALA A MUCHIILOR ARBORELUI MINIM:',
     */,12X,43('-'),/,27X,'VAL=',F10.2) .
   24 STOP
      END



C     *********************************
C     *PROGRAMUL DRUMIN*
C     *********************************
C     DETERMINAREA DRUMULUI DE VALOARE MINIMA INTR-UN GRAF CU
C                   ALGORITMUL BELLMAN-KALABA
C
C
C    LISTA NOTATIILOR
C      N-ORDINUL GRAFULUI
C      M(I,J)-MATRICEA ASOCIATA GRAFULUI
C      V(I,K)-VECTORUL COLOANA CONSTRUIT LA FIECARE ITERATIE K
C      X(I)-SUCCESIUNEA VIRFURILOR DRUMULUI DE VALOARE MINIMA
C      VAL-VALOAREA DRUMULUI MINIM IN GRAF
C
C    ALOCAREA MEMORIEI
       INTEGER M(200,200),V(200,200),X(200)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
       READ(1,1) N
     1 FORMAT (I5)
       DO 2 I=1,N
     2 READ (1,3) (M(I,J),=1,N)
     3 FORMAT (1215)
       WRITE (3,4)
     4 FORMAT (////,24X,32('*'),/,24X,'*PROGRAMUL DRUMIN*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'),
     */,24X,'MATRICEA M(I,J) ATASATA GRAFULUI',/,24X,32('-'))
       DO 5 I=1,N
     5 WRITE (3,6) (M(I,J),J=1,N)
     6 FORMAT (/,13(1X,I4))
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
       WRITE (3,7)
     7 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'),/,22X,
     *'VIRFURILE DRUMULUI DE VALOARE MINIMA',/,22X,36('-'))
       S=0
       DO 8 I=1,N
       DO 8 J=1,N
     8 S=S+M(I,J)
       DO 9 I=1,N
     9 M(I,I)=0
       DO 10 I=2,N
       II=I-1
       DO 10 J=1,II
    10 M(I,J)=S
       DO 12 I=1,N
       I2=I+1
       DO 11 J=I2,N
       IF (M(I,J).NE.0) GOTO 11
       M(I,J)=S
    11 CONTINUE
    12 CONTINUE
       K=1
       J=N
       DO 13 I=1,N
    13 V(I,K)=M(I,J)
       K=K+1
    14 DO 17 I=1,N
       J=1
       IA=K-1
       B=V(J,IA)+M(I,J)
       MIN=B
    15 J=J+1
       IF (J.LE.N) GOTO 16
       B=V(J,IA)+M(I,J)
       IF (MIN.LT.B) GOTO 15
       MIN=B
       GOTO 15
    16 V(I,K)=MIN
    17 CONTINUE
       K=K+1
       IC=K-1
       IA=K-2
       DO 18 J=1,N
       IF (V(J,IA).EQ.V(J,IC)) GOTO 14
    18 CONTINUE
       VAL=V(1,IC)
       J=1
       IC=K-1
       X(1)=1
       ND=1
       DO 19 L=2,N
       D=V(J,IC)-V(L,IC)
       IF (D.NE.M(J,L)) GOTO 19
       ND=ND+1
       X(ND)=L
       J=L
    19 CONTINUE
       WRITE (3,20) (X(I),I=1,ND)
    20 FORMAT (/,6X,15(I3,1X))
       WRITE (3,21) VAL
    21 FORMAT (/,27X,'VALOAREA MINIMA A DRUMULUI',/,27X,26('-'),
     */,35X,'VAL=',I5)
       STOP
       END



C     *********************************
C     *PROGRAMUL OPTIM1*
C     *********************************
C     REZOLVAREA PROBLEMELOR PRACTICE DE CALCUL HIDRAULIC OPTIM
C       AL CANALELOR CU TALUZE PLANE SI RADIER CILINDRIC
C
C
C    LISTA NOTATIILOR
C      N-COEFICIENTUL DE RUGOZITATE
C      M-COEFICIENTUL DE INCLINARE AL TALUZULUI
C      I-PANTA LONGITUDINALA A CANALULUI
C      RO-RAZA ARCULUI DE CERC DE LA RADIER ,IN M
C      HO,BO-INALTINEA SI LATINEA CORESPUNZATOARE PORTIUNII CIRCULARE
C           A SECTIUNII TRANSVERSALE ,IN M
C      R-RAZA HIDRAULICA A SECTIUNII TRANSVERSALE ,IN M
C      H-ADINCIMEA APEI ,IN M
C      B-LATIMEA LA OGLINDA APEI ,IN M
C      Q-DEBITUL ,IN MC/S
C      Y-EXPONENT IN FORMULA PAVLOVSKI
```


Scanned with OKEN Scanner

```
C     EPS-EROAREA ADMISA IN CALCUL
C     KOD-IDENTIFICATOR CARUIA I SE ATRIBUIE VALOAREA 1 CIND
C         SE CUNOASTE Q (W=Q SI T=0),RESPECTIV VALOAREA 0,CIND
C         SE CUNOASTE H (W=H SI T=0) SAU SE CUNOASTE B (W=0 SI T=B)
C     VMIN,VMAX-LIMITA MINIMA SI MAXIMA ADMISIBILA A VITEZEI V A
C               APEI ,IN M/S
C
C   ALOCAREA MEMORIEI
      REAL N,M,I
      EXTERNAL RY
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N,I,M,VMIN,VMAX,W,T,EPS,KOD
    1 FORMAT (2F8.4,3F6.2,2F7.3,F8.4,I2)
      WRITE (3,2)
    2 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  O P T I M 1*',
     */,24X,32('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('*'))
      WRITE (3,3) N,M,I,VMIN,VMAX,EPS,KOD
    3 FORMAT (/,19X,'N=',F6.4,3X,'M=',F4.5,3X,'I=',F6.4,/,6X,'VMIN=',
     *F5.2,' M/S',2X,'VMAX=',F5.2,' M/S',2X,'EPS=',F6.4,2X,'KOD=',I1)
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      IF (KOD.EQ.1) GOTO 9
      IF (T.EQ.0) GOTO 6
      WRITE (3,4) T
    4 FORMAT (/,28X,'B=',F6.2,' M')
      B=T
      CALL PARAM (M,BETO,F1,F2,F,FI,TAU,PSI)
      CALL CERC (F1,F2,FI,H,RO,BO,HO)
      H=B*FI/TAU
      CALL RY (N,F,FI,PSI,H,R,Y)
      CALL QV (N,F,FI,PSI,R,Y,H,VMIN,VMAX,I,Q,V)
      WRITE (3,5) H,Q
    5 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'),
     */,22X,'H=',F6.3,' M',3X,'Q=',F6.3,' MC/S)
      GOTO 17
    6 WRITE (3,7) T
    7 FORMAT (/,28X,'H=',F6.2,' M')
      H=W
      CALL PARAM (M,BETO,F1,F2,F,FI,TAU,PSI,)
      CALL CERC (F1,F2,F,H,RO,BO,HO)
      B=H*TAU/FI
      CALL RY (N,F,FI,PSI,H,R,Y)
      CALL QV (N,F,FI,PSI,R,Y,H,VMIN,VMAX,I,Q,V)
      WRITE (3,8) B,Q
    8 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'),
     */,22X,'B=',F6.3,' M',3X,'Q=',F6.3,' MC/S')
      GOTO 17
    9 WRITE (3,10) W
   10 FORMAT (/,26X,'Q=',F6.3,' MC/S')
      Q=W
      CALL PARAM (M,BETO,F1,F2,F,FI,TAU,PSI)
      H1=0.05
   11 CALL RY (N,F,FI,PSI,H,R,Y)
      H2=F**(Y+0.5)*FI**(Y+2.5)/PSI**(2.*Y+3.)
      H2=(H2*N*Q/SQRT(I))**(1./(Y+2.5))
      IF (ABS(H2-H1).LE.EPS) GOTO 12
      H1=(H1+H2)/2
      GOTO 11
   12 H=(H1+H2)/2
      CALL RY (N,F,FI,PSI,H,R,Y)
      V=R**(Y+0.5)*I**0.5/N
      IF (V.GE.VMIN) GOTO 13
      V=VMIN
      GOTO 14
   13 IF (V.LE.VMAX) GOTO 15
      V=VMAX
   14 CALL HI (N,F,FI,PSI,Q,V,R,Y,H,I,RY)
   15 CALL CERC (F1,F2,FI,H,RO,BO,HO)
      B=H*TAU/FI
      WRITE (3,16) H,B
   16 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('*'),
     */,23X,'H=',F6.3,' M',3X,'B=',F6.3,' M')
   17 WRITE (3,18) BETO,Y,R,RO,HO,V,I
   18 FORMAT (/,20X,'BETO=',F4.2,2X,'Y=',F7.5,2X,'R=',F5.2,' M',/,12X,
     *'RO=',F6.3,' M',2X,'HO=',F6.3,' M',2X,'V=',F6.3,' M/S',2X,'I=',F6.4)
      STOP
      END

      SUBROUTINE PARAM (M,BETO,F1,F2,F,FI,TAU,PSI)
      REAL M
      BETO=2*SQRT(1+M**2)
      IF (BETO.LE.2*M) GOTO 1
      IF (BETO.LT.4/(BETO-2*M)) GOTO 3
    1 WRITE (3,2)
    2 FORMAT (//,31X,'INCOMPATIBILITATE')
      STOP
    3 F1=4/BETO
      F2=1-2*M/BETO
      F3=ATAN(1/M)
      F4=F3-2*M/BETO
      ALFA=(F1-BETO*F2)/(BETO-2*M)
      FI=ALFA+F2
      TAU=2*M*ALFA+F1
      PSI=SQRT(M*ALFA**2+F1*ALFA+F4)
      F=BETO*ALFA*2+F3
      RETURN
      END

      SUBROUTINE CERC (F1,F2,FI,H,RO,BO,HO)
      RO=H/FI
      BO=F1*RO
      HO=F2*RO
      RETURN
      END

      SUBROUTINE RY (N,F,FI,PSI,H,R,Y)
      REAL N
      R=H*PSI**2/(F*FI)
      Y=2.5*SQRT(N)-0.13-0.75*SQRT(R)*(SQRT(N)-0.10)
      RETURN
      END

      SUBROUTINE HI (N,F,FI,PSI,Q,V,R,Y,H,I,RY)
      REAL N,I
      H=FI/PSI*SQRT(Q/V)
      CALL RY (N,F,FI,PSI,H,R,Y)
      I=N**2*V**2/R**(2*Y+1)
      RETURN
      END

      SUBROUTINE QV (N,F,FI,PSI,R,Y,H,VMIN,VMAX,I,Q,V)
      REAL N,I
      V=R**(Y+0.5)*I**0.5/N
      IF (V.GE.VMIN) GOTO 1
      V=VMIN
      IF (V.LE.VMAX) GOTO 2
      V=VMAX
    1 I=N**2*V**2/R**(2*Y+1)
    2 Q=PSI**(2*Y+3)/(N*F**(Y+2.5)*F**(Y+0.5))
      Q=Q*H**(Y+2.5)*I**0.5
      RETURN
      END



C     ************************************
C     * P R O G R A M U L  O T R E D I R A *
C     ************************************
C     DETERMINAREA TRASEELOR OPTIME ALE UNEI RETELE RAMIFICATE
C                 (ARBORII MINIMI AI UNUI GRAF)
C
C
C   LISTA NOTATIILOR
C     N-ORDINUL GRAFULUI ATASAT RETELEI
C     C(I,J)-MATRICEA CRITERIALA ATASATA GRAFULUI
C     MIN(I)-MINIMUL ELEMENTELOR DE PE LINIA I
C     NEM(I)-NUMARUL ELEMENTELOR MINIME DE PE LINIA I
C     MAR(I)-NUMARUL ELEMENTELOR MARCATE DE PE LINIA I
```


Scanned with OKEN Scanner

```
C     M(L,K)-MATRICEA VIRFURILOR MUCHIILOR OPTIME SI A VALORILOR
C           CORESPUNZATOARE
C     R,S-INDICII ELEMENTULUI MINIM
C     MINLM-MINIMUL ELEMENTELOR MARCATE DE PE LINIILE CE CONTIN
C           ELEMENTELE MARCATE
C     F(I)-FRECVENTA ABSOLUTA A CELOR N-1 VIRFURI ALE GRAFULUI
C         SITUATE PE COLOANA 2 ORDONATA A MATRICEI M(K,L)
C     FF(I)-FRECVENTA CUMULATA A CELOR N-1 VIRFURI ALE GRAFULUI
C           DE FRECVENTA ABSOLUTA F(I)
C     NN(I,K)-BLOCUL ELEMENTELOR MULTIMII REZULTATE DIN INTERSECTIA
C            INTERVALULUI INCHIS LA DREAPTA (FF(I-1),FF(I)) CU MUL-
C            TIMEA NUMERELOR NATURALE
C     A(I,J)-MATRICEA PE A CAREI COLOANE SE GASESC INDICII LINIILOR
C           MATRICEI ORDONATE M(K,L),CE CONTIN MUCHIILE UNUIA
C           DIN ARBORII OPTIMI
C
C    ALOCAREA MEMORIEI
      INTEGER C(200,200),M(200,3),MIN(200),NEM(200),MAR(200),
     *F(200),FF(200),NN(200,40),KS(200),A(200,25),R,S
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) N
    1 FORMAT (I5)
      DO 2 I=1,N
    2 READ (1,3) (C(I,J),J=1,N)
    3 FORMAT (12I5)
      WRITE (3,4)
    4 FORMAT (////,22X,36('*'),/,22X,'*P R O G R A M U L  O T R E D I R A*',
     */,22X,36('*'),//,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,30('='),
     *//,23X,'MATRICEA VALORILOR MUCHIILOR C(I,J)',/,23X,35('*'),/)
      DO 5 I=1,N
    5 WRITE (3,6) (C(I,J),J=1,N)
    6 FORMAT (/,6X,12(I4,1X))
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,7)
    7 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('='))
      DO 9 I=1,N
      DO 9 J=1,N
      IF (J.EQ.1) GOTO 8
      IF (MIN(I).LT.C(I,J)) GOTO 9
    8 MIN(I)=C(I,J)
    9 CONTINUE
      DO 10 I=1,N
      NEM(I)=0
      MAR(I)=0
      DO 10 J=1,N
      IF (C(I,J).EQ.MIN(I)) NEM(I)=NEM(I)+1
   10 CONTINUE
      I=1
   11 IF (NEM(I).EQ.1) GOTO 13
      I=I+1
      IF (I.LE.N) GOTO 11
      WRITE (3,12)
   12 FORMAT (//,24X,'NU EXISTA LINIE CU ELEMENT UNIC')
      GOTO 65
   13 DO 15 J=1,N
      IF (J.EQ.1) GOTO 14
      IF (MIN(I).LT.C(I,J)) GOTO 15
   14 MIN(I)=C(I,J)
      R=I
      S=J
   15 CONTINUE
      K=1
      KK=K
      L=1
      M(K,L)=R
      L=L+1
      M(K,L)=S
      L=L+1
      M(K,L)=MIN(I)
      C(R,S)=9998
      C(S,R)=9998
      MAR(R)=MAR(R)+1
      MAR(S)=MAR(S)+1
   16 I=1
   17 IF (MAR(I).NE.0) GOTO 19
      I=I+1
      IF (I.LE.N) GOTO 17
      WRITE (3,18)
   18 FORMAT (///,37X,'EROARE')
      GOTO 65
   19 DO 21 J=1,N
      IF (J.EQ.1) GOTO 20
      IF (MINLM.LT.C(I,J)) GOTO 21
   20 MINLM=C(I,J)
      R=I
      S=J
   21 CONTINUE
   22 I=I+1
      IF (I.GT.N) GOTO 24
      IF (MAR(I).EQ.0) GOTO 22
      DO 23 J=1,N
      IF (MINLM.LT.C(I,J)) GOTO 23
      MINLM=C(I,J)
      R=I
      S=J
   23 CONTINUE
      GOTO 22
   24 IF (MAR(S).NE.0) GOTO 26
      DO 25 I=1,N
      IF (MAR(I).EQ.0) GOTO 25
      IF (C(I,S).NE.C(R,S) GOTO 25
      K=K+1
      KK=K
      L=1
      M(K,L)=I
      L=L+1
      M(K,L)=S
      L=L+1
      M(K,L)=C(I,S)
      C(I,S)=9998
      C(S,I)=9998
      MAR(I)=MAR(I)+1
      MAR(S)=MAR(S)+1
   25 CONTINUE
      GOTO 26
   26 C(R,S)=9998
      C(S,R)=9998
      MAR(R)=MAR(R)+1
      MAR(S)=MAR(S)+1
   27 DO 28 I=1,N
      IF (MAR(I).EQ.0) GOTO 16
   28 CONTINUE
      WRITE (3,29)
   29 FORMAT (//,16X,'MATRICEA VIRFURILOR MUCHIILOR OPTIME SI ',
     */,16X,39('*'),/,17X,'A VALORILOR CORESPUNZATOARE M(K,L)',
     */,17X,34('*'),/)
      DO 31 K=1,KK
      WRITE (3,30) (M(K,L),L=1,3)
   30 FORMAT (23X,3(I4,3X),/)
   31 CONTINUE
      KK1=KK-1
      DO 34 K=1,KK1
   32 IF (M(K,2).LE.M(K+1,2)) GOTO 34
      DO 33 L=1,3
      M1=M(K,L)
      M(K,L)=M(K+1,L)
   33 M(K+1,L)=M1
      IF (K.EQ.1) GOTO 34
      K=K-1
      GOTO 32
   34 CONTINUE
      WRITE (3,35)
   35 FORMAT (///,16X,'MATRICEA M(K,L) CU COLOANA 2 ORDONATA',/,16X,37('*'),/)
      DO 36 K=1,KK
   36 WRITE (3,30) (M(K,L),L=1,3)
      N1=N-1
      DO 37 I=1,N1
   37 F(I)=1
      I=0
      K=1
```


Scanned with OKEN Scanner

```
   38 IF (M(K,2).EQ.M(K+1,2)) GOTO 41
   39 I=I+1
      F(I)=F(I)
   40 K=K+1
      IF (K.LE.KK1) GOTO 38
      GOTO 44
   41 I=I+1
      F(I)=F(I)+1
   42 K=K+1
      IF (M(K,2).EQ.M(K+1,2)) GOTO 43
      IF (K.EQ.KK1) GOTO 39
      GOTO 40
   43 F(I)=F(I)+1
      IF (K.LT.KK1) GOTO 42
   44 NA=1
      DO 47 I=1,N1
      IF (I.EQ.1) GOTO 45
      FF(I)=FF(I-1)+F(I)
      GOTO 46
   45 FF(I)=F(I)
   46 NA=NA+F(I)
   47 CONTINUE
      WRITE (3,48) NA
   48 FORMAT (//,28X,'NUMARUL DE ARBORI MINIMI',/,28X,24('*'),
     */,36X,'NA=',I5)
      DO 51 I=1,N1
      K=1
      IF (I.EQ.1) GOTO 50
      NN(I,K)=FF(I-1)+1
   49 KS(I)=K
      IF (NN(I,K).GE.FF(I)) GOTO 51
      K=K+1
      NN(I,K)=NN(I,K-1)+1
      GOTO 49
   50 NN(I,K)=1
      GOTO 49
   51 CONTINUE
      DO 53 I=1,N1
      N2=NA/F(I)
      J=1
      K=1
   52 A(I,J)=NN(I,K)
      J=J+1
      IF (J.LE.N2) GOTO 52
      IF (J.GT.NA) GOTO 53
      N2=N2+NA/F(I)
      K=K+1
      GOTO 52
   53 CONTINUE
      I=1
      J=1
   54 IF (A(I,J).NE.A(I,J+1)) GOTO 60
      I=I+1
      IF (I.LE.N1) GOTO 54
   55 I=I-1
      IF (I.LT.1) GOTO 58
      L=J+2
   56 IF (L.GT.NA) GOTO 55
      IF (A(I,J+1).NE.A(I,L)) GOTO 57
      L=L+1
      GOTO 56
   57 IS=A(I,J+1)
      A(I,J+1)=A(I,L)
      A(I,L)=IS
      GOTO 60
   58 WRITE (3,59)
   59 FORMAT (//,10X,'MATRICEA A(I,J) NU POATE FI ORDONATA LEXICOGRAFIC')
      GOTO 65
   60 J=J+1
      IF (J.GT.NA-1) GOTO 61
      I=1
      GOTO 54
   61 WRITE (3,62)
   62 FORMAT (//,10X,'MATRICEA A(I,J) CU COLOANELE ORDONATE  ,
     *' LEXICOGRAFIC',/,10X,50('*'),/)
      DO 63 I=1,N1
      WRITE (3,64) (A(I,J),J=1,NA)
   64 FORMAT (6X,10(2X,I4),/)
   65 STOP
      END
C     ********************************
C     * P R O G R A M U L   A C I R E V *
C     ********************************
C     ANALIZA CICLICA A RETELELOR INELARE DE DISTRIBUTIE A APEI
C                    IN FORMULARE VARIATIONALA
C
C
C   LISTA NOTATIILOR
C      NZ       -NUMARUL ZONELOR DE CONSUM
C      NC       -NUMARUL TRONSOANELOR DE CONDUCTE
C      NI       -NUMARUL INELELOR INDEFENDENTE
C      NCR      -NUMARUL TRONSOANELOR RETELEI VIRTUAL RAMIFICATE
C      NNI      -NUMARUL MAXIM DE NODURI ALE UNUI INEL
C      NS       -NUMARUL SURSELOR DE ALIMENTARE
C      NIF      -NUMARUL INELELOR FICTIVE (NS-1)
C      DELTA    -RUGOZITATEA ABSOLUTA A CONDUCTELOR, IN MM
C      BETA     -EXPONENTUL DEBITULUI IN RELATIA FUNCTIONALA
C                DE FORMA GENERALA
C      A,B,C    -COEFICIENTII FUNCTIEI DE INTERPOLARE A INALTIMII DE
C                POMPARE A POMPELOR INTEGRATE PE ARTERE
C      NO       -NOD OBLIGAT, CU COTA PIEZOMETRICA CUNOSCUTA
C      ZO       -COTA PIEZOMETRICA A NODULUI NO, IN M
C      EST      -VALOAREA ESTIMATIVA A MINIMULUI FUNCTIEI FE(X)
C      EPS      -PRECIZIA CU CARE SE CALCULEAZA MINIMUL
C      LIM      -NUMARUL LIMITA ADMIS AL APELURILOR SUBRUTINEI FUNCT
C      NIT      -NUMARUL ITERATIILOR PARCURSE
C      KOD      -IDENTIFICATOR CU VALORILE : 0 SAU 1-DUPA CUM S-A
C                OBTINUT SAU NU MINIMUL INTR-UN NUMAR DE APELURI
C                ALE "FUNCT" MAI MIC SAU EGAL CU LIM ; 2-CIND NU
C                EXISTA MINIM ; 3-CIND EXISTA ERORI IN CALCULUL
C                COMPONENTELOR GRADIENTULUI
C      NIN(I)   -NODUL INITIAL AL TRONSONULUI I
C      NFI(I)   -NODUL FINAL AL TRONSONULUI I
C      L(I)     -LUNGIMEA TRONSONULUI I, IN M
C      D(I)     -DIAMETRUL TRONSONULUI I, IN M
C      NP(I)    -NUMARUL POMPELOR INTEGRATE PE ARTERA I
C      S(I)     -MODULUL DE REZISTENTA HIDRAULICA AL TRONSONULUI I,
C                IN S=*2/M**5
C      Q(I)     -DEBITUL TRANSPORTAT PE TRONSONUL I, IN MC/S
C      H(I)     -PIERDEREA DE SARCINA PE TRONSONUL I, IN M
C      V(I)     -VITEZA APEI IN TRONSONUL I, IN M/S
C      QC(N)    -DEBITUL DE CONSUM CONCENTRAT, SAU DE ALIMENTARE
C                IN NODUL N, IN MC/S
C      ZT(N)    -COTA GEODEZICA A NODULUI N, IN M
C      Z(N)     -COTA PIEZOMETRICA IN NODUL N, IN M
C      HD(N)    -PRESIUNEA DISPONIBILA IN NODUL N, IN M
C      QO(KZ)   -DEBITUL DE ALIMENTARE AL RETELEI, CORESPUNZATOR
C                ZONEI DE CONSUM KZ, IN MC/S
C      AL(I,KZ) -LUNGIMEA TRONSONULUI I AFLAT IN ZONA DE CONSUM
C                KZ, IN M
C      IN(JN,JM)-MATRICEA CUPRINZIND NODURILE JN ALE INELULUI JM
C      ZI(JM)   -COTA PIEZOMETRICA IN NODUL DE INTRARE IN CICLUL
C                FICTIV JM
C      ZE(JM)   -COTA PIEZOMETRICA IN NODUL DE IESIRE DIN CICLUL
C                     FICTIV JM
C      X(JN)    -CORECTIA DEBITULUI IN INELUL JM, IN MC/S
C
C  ALOCAREA MEMORIEI
      DIMENSION AL(300,5),QO(5),Q(300),D(300),S(300),H(300),V(300),
     *QC(230),ZT(230),Z(230),HD(230),QD(230),QSP(5),X(80),
     *ZI(10),ZE(10)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      EXTERNAL FUNCT,ORIENT
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,2) NZ,NC,NCR,NI,NNI,NS,LIM,DELTA,BETA,A,B,C,NO,ZO,
     *EST,EPS
```


Scanned with OKEN Scanner

```
    2 FORMAT (715,2F5.2,F10.2,2F8.2,/,I5,F8.2,F10.4,F10.8)
      READ (1,4) ((AL(I,KZ),KZ=1,NZ),I=1,NC)
    4 FORMAT (10F6.1)
      READ (1,6) (QO(KZ),KZ=1,NZ)
    6 FORMAT (8F8,3)
      NCR1=NCR+1
      READ (1,8) (QC(N),ZT(N),N=1,NCR1)
    8 FORMAT (3(F8.5,F8.2))
      DO 11 I=1,NC
      READ (1,10) NIN(I),NFI(I),NP(I),D(I)
   10 FORMAT (3I5,F5.3)
   11 CONTINUE
      IF (NC.EQ.NCR) GO TO 16
      DO 15 JM=1,NI
      READ (1,13) (IN(JN,JM),JN=1,NN1)
   13 FORMAT (15I5)
   15 CONTINUE
      NIF=NS-1
      IF (NIF.EQ.0) GO TO 16
      DO 120 K=1,NIF
      READ (1,6) ZI(K),ZE(K)
  120 CONTINUE
   16 WRITE (3,17)
   17 FORMAT (////,24X,32('*'),/,24X,'*P R O G R A M U L  '
     *'A C I R E V*',/,24X,32('*'))
      WRITE (3,19)
   19 FORMAT (////,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,
     *30('*'),///,34X,'DATE GENERALE',/,34X,13('='))
      WRITE (3,20) NZ,NC,NCR,NI,NN1,NS,LIM,DELTA,BETA,A,B,C,NO,ZO,
     *EST,EPS
   20 FORMAT (/,3X,'NZ=',I2,2X,'NC=',I3,2X,'NCR=',I3,2X,
     *'NI=',I3,2X,'NN1=',I2,2X,'NS=',I2,2X,'LIM=',I4,/,7X,'DELTA=',
     *F4.2,' MM',2X,'BETA=',F4.2,2X,'A=',F9.2,2X,'B=',F7.2,2X,'C=',F5.2,
     * /,1X,'NO=',I3,2X,'ZO=',F6.2,' M',2X,'EST=',E14.6,2X,
     *'EPS=',E15.8)
      WRITE (3,21)
   21 FORMAT (//.21X,'DEBITUL DE ALIMENTARE PE ZONE DE CONSUM',
     */,21X,39('='),//,32X,'KZ',3X,'QO(KZ), MC/S',/,30X,19('-'))
      DO 24 KZ=1,NZ
      WRITE (3,22) KZ,QO(KZ)
   22 FORMAT (/,30X,I4,3X,F8.5)
      IF (KZ.NE.NZ) GO TO 24
      WRITE (3,23)
   23 FORMAT (30X,19('-'))
   24 CONTINUE
      WRITE (3,25)
   25 FORMAT (//,25X,'DEBITELE CONCENTRATE IN NODURI',/,
     *25X,30('='))
      WRITE (3,26) (QC(N),N=1,NCR1)
   26 FORMAT (/,4X,7F8.5)
      WRITE (3,27)
   27 FORMAT (//,27X,'COTELE GEODEZICE IN NODURI',/,
     *27X,26('='))
      WRITE (3,28) (ZT(N),N=1,NCR1)
   28 FORMAT (/,4X,7F8.2)
      WRITE (3,29)
   29 FORMAT (//,25X,'LUNGIMEA TRONSOANELOR PE ZONE',/,
     *25X,29('='))
      WRITE (3,30) (( AL(I,KZ),KZ=1,NZ),I=1,NC)
   30 FORMAT (/,4X,7F8.1)
      IF (NC.EQ.NCR) GO TO 36
      WRITE (3,31)
   31 FORMAT (//,31X,'MATRICEA INELELOR',/,31X,17('='))
      DO 33 JM=1,NI
      WRITE (3,32) (IN(JM,JM),JN=1,NN1)
   32 FORMAT (/,5X,1115)
   33 CONTINUE
      IF (NIF.EQ.0) GO TO 36
      WRITE (3,130)
  130 FORMAT (//,25X,'COTELE PIEZOMETRICE LA INTRARE',
     */,25X,30('='),/25X,'SI IESIRE DIN CICLURILE FICTIVE',
     */,25X,31('='))
      WRITE (3,135) (ZI(JM),ZE(JM),JM=1,NIF)
  135 FORMAT (/,4X,8F8.2)
C
C  EFECTUAREA CALCULULUI SI SCRIEREA REZULTATELOR
   36 WRITE (3,37)
   37 FORMAT(///,29X,'REZULTATELE CALCULULUI',/,29X,22('='))
      DO 39 I=1,NC
      L(I)=0
      DO 38 KZ=1,NZ
      L(I)=L(I)+AL(I,KZ)
   38 CONTINUE
   39 CONTINUE
      CALL REDEC (AL,QO,QC,QSP,QD,Q)
      WRITE (3,40)
   40 FORMAT (//,22X,'DEBITELE SPECIFICE PE ZONE DE CONSUM',
     */,22X,36('='),//,29X,24('='),/,31X,'KZ',7X,'QSP(KZ), MC/S',/,29X,
     *24('-'))
      DO 43 KZ=1,NZ
      WRITE (3,41) KZ,QSP(KZ)
   41 FORMAT (/,29X,I4,6X,F10.8)
      IF (KZ.NE.NZ) GO TO 43
      WRITE (3,42)
   42 FORMAT (29X,24('-'))
   43 CONTINUE
      WRITE (3,44)
   44 FORMAT (//,25X,'DEBITELE DISTRIBUITE IN NODURI',/,
     *25X,30('='))
      WRITE (3,45) (QD(N),N=1,NCR1)
   45 FORMAT (/,4X,7F8.5)
      IF (NC.EQ.NCR) GOTO 51
      CALL OPREDEC (IO,NIF,Q,ORIENT)
      WRITE (3,46)
   46 FORMAT (//,15X,'LUNGIMILE,DIAMETRELE SI DEBITELE',
     * ' PE TRONSOANE',/,15X,48('='),//,1X,59('-'),/,4X,
     *'I',3X,'NIN(I)',2X,'NFI(I)',3X,'D(I), M',2X,'L(I), M',
     *2X,'Q(I), MC/S',2X,'NP(I)',/,1X,59('-'))
      DO 49 I=1,NC
      WRITE (3,47) I,NIN(I),NFI(I),D(I),L(I),Q(I),NP(I)
   47 FORMAT (/,1X,I4,2(4X,I4),5X,F5.3,4X,F5.1,4X,F8.5,4X,I2)
      IF (I.NE.NC) GO TO 49
      WRITE (3,48)
   48 FORMAT (1X,59('-'))
   49 CONTINUE
C   DETERMINAREA UNEI APROXIMARI INITIALE A CORECTIILOR
C   DEBITELOR PE INELE
      DO 50 I=1,NC
      CALL CLAM (I,DELTA,Q,D,BETA,S)
   50 CONTINUE
      CALL COREC (IO,NIF,ZI,ZE,A,B,C,S,Q,X)
C   APLICAREA METODEI GRADIENTILOR CONJUGATI LA
C   ECHILIBRAREA DEBITELOR
      CALL GRACON (NI,NIF,IO,ZI,ZE,A,B,C,S,Q,X,FUNCT)
C   CALCULUL CARACTERISTICILOR HIDRAULICE
   51 DO 52 I=1,NC
      CALL CLAM (I,DELTA,Q,D,BETA,S)
   52 CONTINUE
      CALL PIERDS (Q,S,H)
      CALL VIT(Q,D,V)
      WRITE(3,53)
   53 FORMAT(//,15X,'CARACTERISTICILE HIDRAULICE DUPA ECHILIBRARE',
     */,15X,44('='),//,1X,61('-'),/,3X,'I',2X,'D(I), M',
     *2X,'Q(I), MC/S',2X,'S(I), S**2/N**5',2X,'H(I), M',
     *2X,'V(I), M/S',/,1X,61('-'))
      DO 56 I=1,NC
      WRITE (3,54) I,D(I),Q(I),S(I),H(I),V(I)
   54 FORMAT (/,1X,I3,3X,F5.3,4X,F8.5,5X,F9.1,5X,F5.2,5X,F7.2)
      IF (I.NE.NC) GOTO 56
      WRITE (3,55)
   55 FORMAT(1X,61('-'))
   56 CONTINUE
      DO 57 K=1,NCR1
   57 Z(K)=0.
      Z(NO)=ZO
      HD(NO)=Z(NO)-ZT(NO)
   58 DO 62 I=1,NC
      N1=NIN(I)
      N2=NFI(I)
      KO=1
      IF (Z(N1).NE.0.AND.Z(N2).EQ.0) GOTO 60
      IF (Z(N1).EQ.0.AND.Z(N2).NE.0) GOTO 59
      GOTO 62
```


Scanned with OKEN Scanner

```
   59 N3=N1
      N1=N2
      N2=N3
      K0=-1
   60 Z(N2)=Z(N1)-K0*H(I)
      IF (NP(I).EQ.0) GOTO 61
      HP=A*ABS(Q(I))**2/NP(I)**2+B*ABS(Q(I))/NP(I)+C
      Z(N2)=Z(N2)+K0*HP*Q(I)/ABS(Q(I))
   61 HD(N2)=Z(N2)-ZT(N2)
   62 CONTINUE
      DO 63 I=1,NC
      N1=NIN(I)
      N2=NFI(I)
      IF (Z(N1).NE.0.AND.Z(N2).NE.0) GOTO 63
      GOTO 58
   63 CONTINUE
      WRITE (3,64)
   64 FORMAT(//,25X,'COTELE PIEZOMETRICE NODALE',/,
     *25X,26('='),//,10X,47('-'),/,12X,'I',2X,'NIN',2X,'NFI',
     *2X,'ZIN, M',3X,'ZFI, M',3X,'HDI, M',2X,'HDF, M',/,10X,47('-'))
      DO 67 I=1,NC
      N1=NIN(I)
      N2=NFI(I)
      WRITE (3,65) I,NIN(I),NFI(I),Z(N1),Z(N2),HD(N1),HD(N2)
   65 FORMAT (/,8X,3(2X,I3),2(2X,F7.3),2(2X,F6.3))
      IF (I.NE.NC) GOTO 67
      WRITE (3,66)
   66 FORMAT (10X,47('-'))
   67 CONTINUE
      STOP
      END

      SUBROUTINE REDEC (AL,QO,QO,QSP,QD,Q)
C   REPARTITIA DEBITELOR DE CALCUL
      DIMENSION AL(300,5),QO(5),QC(230),QSP(5),Q(300),SAL(300,5),
     *QD(230),QK(230)
      INTEGER NIN(300),NFI(300),IN(15,80),NP(300)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN.NCR,NNI,NP,EST,EPS,LIM
      DO 2 KZ=1,NZ
      SAL(1,KZ)=AL(1,KZ)
      DO 1 I=2,NC
    1 SAL(I,KZ)=SAL(I-1,KZ)+AL(I,KZ)
      QSP(KZ)=QO(KZ)/SAL(NC,KZ)
    2 CONTINUE
      NCR1=NCR+1
      DO 3 I=1,NCR1
      QD(I)=0
      QK(I)=0
    3 CONTINUE
      DO 5 I=L,NC
      DO 4 KZ=1,NZ
      I1=NIN(I)
      I2=NFI(I)
      QD(I1)=QD(I1)+0.5*QSP(KZ)*AL(I,KZ)
      QD(I2)=QD(I2)+0.5*QSP(KZ)*AL(I,KZ)
    4 CONTINUE
    5 CONTINUE
      DO 6 KZ=1,NZ
    6 QD(NCR)=QD(NCR)+0.5*QSP(KZ)*AL(NCR,KZ)
      QD(NCR+1)=0
      DO 7 I=1,NC
    7 Q(I)=0
      DO 8 I=1,NCR
      QK(I)=QK(I)+QD(I)-QC(I)
      Q(I)=QK(I)
      I3=NIN(I)
      QK(I3)=QK(I3)+Q(I)
    8 CONTINUE
      RETURN
      END

      SUBROUTINE OPREDEC (IO,NIF,Q,ORIENT)
C   OPTIMIZAREA REPARTITIEI DEBITELOR DE CALCUL
      DIMENSION Q(300),DQ(80)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      CALL ORIENT(IO)
      NII=NI-NIF
      ITER=0
    1 ITER=ITER+1
      DO 4 JM=1,NII
      QQ=0.
      DD=0.
      DO 2 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 3
      K=IABS(IO(JM,JN))
      DD=DD+L(K)
      QQ=QQ+L(K)*Q(K)*IO(JM,JN)/K
    2 CONTINUE
    3 DQ(JM)=-QQ/DD
    4 CONTINUE
      DO 7 JM=1,NII
      DO 5 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 6
      K=IABS(IO(JM,JN))
      Q(K)=Q(K)+DQ(JM)*IO(JM,JN)/K
    5 CONTINUE
    6 CONTINUE
    7 CONTINUE
      IF (ITER.GT.200) GOTO 9
      DO 8 JM=1,NII
      IF (ABS(DQ(JM)).GT.0.0001) GOTO 1
    8 CONTINUE
    9 CONTINUE
      RETURN
      END

      SUBROUTINE ORIENT (IO)
C   DETERMINAREA ORIENTARII TRONSOANELOR
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      DO 6 JM=1,NI
      DO 1 JN=1,NNI
    1 IO(JM,JN)=0
      DO 5 JN=1,NNI
      N1=IN(JN,JM)
      N2=IN(JN+1,JM)
      IF (N2.EQ.0) GOTO 6
      DO 2 I=1,NC
      IF (N1.EQ.NIN(I).AND.N2.EQ.NFI(I)) GOTO 3
      IF (N1.EQ.NFI(I).AND.N2.EQ.NIN(I)) GOTO 4
    2 CONTINUE
    3 IO(JM,JN)=I
      GOTO 5
    4 IO(JM,JN)=-I
    5 CONTINUE
    6 CONTINUE
      RETURN
      END

      SUBROUTINE CLAM (I,DELTA,Q,D,BETA,S)
C   CALCULUL COEFICIENTULUI LAMBDA SI A REZISTENTEI HIDRAULICE S
      DIMENSION Q(300),D(300),S(300)
      INTEGER NIN(300),NFI(300),NP(300).IN(15,80)
      REAL LAMBDA,L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EFS,LIM
      RALP=1/(-2*ALOG10(DELTA/(1000*D(I)))+1.138)
      IF (BETA.NE.2) GOTO 1
      LAMBDA=RALP**2
      GOTO 2
    1 A2=DELTA*ABS(Q(I))/(1.13*D(I)**2)
      A1=(1274*A2+8)*RALP-4
      LAMBDA=((A1+SQRT((ABS(A1))**2+20380*A2*RALP))/(2550*A2))**2
    2 S(I)=0.0826*LAMBDA*L(I)/D(I)**5
      RETURN
      END

      SUBROUTINE COREC (IO,NIF,ZI,ZE,A,B,C,S,Q,DQ)
C   CALCULUL CORECTIILOR INITIALE PE INELE
      DIMENSION Q(300),DQ(80),ZI(10),ZE(10),S(300),DH(80),SR(80)
      INTEGER NIN(300),NFI(300),IN(15,80),IO(80,15),NP(300)
```


Scanned with OKEN Scanner

```
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      DO 5 JM=1,NI
      SR(JM)=0
      DH(JM)=0
      DO 2 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GO TO 2
      I=IABS(IO(JM,JN))
      SR(JM)=SR(JM)+2*S(I)*ABS(Q(I))
      DH(JM)=DH(JM)+S(I)*Q(I)*ABS(Q(I))*IO(JM,JN)/I
      IF (NP(I).EQ.0) GO TO 2
      SR(JM)=SR(JM)-(2*A*ABS(Q(I))/NP(I)+B)
      HP=(A*(ABS(Q(I)))**2)/NP(I)**2+B*ABS(Q(I))/NP(I)+C)
      IF(IO(JM,JN).LT.0) GO TO 1
      DH(JM)=DH(JM)-HP*Q(I)/ABS(Q(I))
      GOTO 2
    1 DH(JM)=DH(JM)+HP*Q(I)/ABS(Q(I))
    2 CONTINUE
      IF ((NI-JM).GT.NIF) GOTO 4
      DO 3 K=1,NIF
      IF (JM.NE.(NI-NIF+K)) GOTO 3
      DH(JM)=DH(JM)+ZE(K)-ZI(K)
    3 CONTINUE
    4 DQ(JM)=-DH(JM)/SR(JM)
    5 CONTINUE
      RETURN
      END

      SUBROUTINE GRACON (N,NIF,IO,ZI,ZE,A,B,C,S,Q,X,FUNCT)
C   APLICAREA METODEI GRADIENTILOR CONJUGATI LA ECHILIBRAREA
C   DEBITELOR
      DIMENSION X(80),G(80),Q(300),D(150),S(300),ZI(10),ZE(10)
      INTEGER IN(15,80),NIN(300),NFI(300),IO(80,15),NP(300)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      NEF=0
      CALL FUNCT (NIF,IO,ZI,ZE,A,B,C,S,Q,X,F,G)
      NEF=NEF+1
      NIT=0
      KOD=0
      N1=N+1
    6 DO 32 LL=1,NI
      NIT=NIT+1
      FY=F
      GP=0.
      DO 7 J=1,N
    7 GP=GP+G(J)*G(J)
      IF (GP.EQ.0.) GOTO 35
      IF (LL.NE.1) GOTO 9
      DO 8 J=1,N
    8 D(J)=-G(J)
      GOTO 11
    9 BETA=GP/GPV
      DO 10 J=1,N
   10 D(J)=BETA*D(J)-G(J)
   11 DY=0.
      SN=0.
      DO 12 J=1,N
      K=J+N
      D(K)=X(J)
      SN=SN+ABS(D(J))
   12 DY=DY+D(J)*G(J)
      IF (DY.GE.0.) GOTO 31
      SNR=1./SN
      FY=F
      PAS=2.*(EST-F)/DY
      R=SNR
      IF (PAS.LE.0.) GOTO 13
      IF (PAS.GE.R) GOTO 13
      R=PAS
   13 PAS=0.
   14 FX=FY
      DX=DY
      DO 15 I=1,N
   15 X(I)=X(I)+R*D(I)
      CALL FUNCT (NIF,IO,ZI,ZE,A,B,C,S,Q,X,F,G)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 33
      FY=F
      DY=0.
      DO 16 I=1,N
   16 DY=DY+G(I)*D(I)
      IF (DY) 17,26,19
   17 IF (FY.GE.FX) GOTO 19
      R=R+PAS
      PAS=R
      IF (SN*R.LE.1*E20) GOTO 14
   18 KOD=2
      GOTO 39
   19 T=0.
   20 IF (R.EQ.0.) GOTO 28
      Y=3.*(FX-FY)/R+DX+DY
      IF (Y**2.LT.DX*DY) GOTO 36
      W=SQRT(Y**2-DX*DY)
      PAS=(DY+W-Y)*R/(DY-DX+2*W)
      DO 21 I=1,N
   21 X(I)=X(I)+(T-PAS)*D(I)
      CALL FUNCT (NIF,IO,ZI,ZE,A,B,C,S,Q,X,F,G)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 33
      IF (F.GT.FX) GOTO 22
      IF (F.LE.FY) GOTO 28
   22 DPAS=0.
      DO 23 I=1,N
   23 DPAS=DPAS+G(I)*D(I)
      IF (DPAS.GE.0.) GOTO 26
      IF (F-FX) 25,24,26
   24 IF (DX.EQ.DPAS) GOTO 28
   25 FX=F
      DX=DPAS
      T=PAS
      R=PAS
      GOTO 20
   26 IF (FY.NE.F) GOTO 27
      IF (DY.EQ.DPAS) GOTO 28
   27 FY=F
      DY=DPAS
      R=R-PAS
      GOTO 20
   28 T=0.
      DO 29 J=1,N
      K=J+N
      D(K)=X(J)-D(K)
   29 T=T+ABS(D(K))
      IF (NIT.LT.N1) GOTO 30
      IF (T.LE.EPS) GOTO 34
   30 IF (FY-F+EPS) 18,38,31
   31 GPV=GP
   32 KOD=0
      GOTO 6
   33 KOD=1
      GOTO 39
   34 IF (GP.GT.EPS) GOTO 38
   35 KOD=0
      GOTO 39
   36 DO 37 J=1,N
      K=N+J
   37 X(J)=D(K)
      CALL FUNCT (NIF,IO,ZI,ZE,A,B,C,S,Q,X,F,G)
      NEF=NEF+1
      IF (NEF.GE.LIM) GOTO 33
   38 IF (KOD.NE.0) GOTO 39
      KOD=3
      GOTO 6
   39 WRITE (3,40) KOD,NIT,F,(X(I),I=1,N)
   40 FORMAT (///,29X,'KOD=',I1,5X,'ITER=',I4,//,14X,'FEMIN=',E16.8,
     */,23X,'VALORILE CORECTIILOR PENTRU FEMIN :',/,4(2X,E16.8))
      DO 42 JM=1,NI
      DO 42 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 41
      I=IABS(IO(JM,JN))
      Q(I)=Q(I)+X(JM)*(IO(JM,JN)/I
   41 CONTINUE
```


Scanned with OKEN Scanner

```
   42 CONTINUE
      RETURN

      END

      SUBROUTINE FUNCT (NIF,IO,ZI,ZE,A,B,C,S,Q,X,FE,G)
C     CALCULUL VALORII FUNCTIEI DE MINIMIZAT SI A COMPONENTELOR
C     GRADIENTULUI
      DIMENSION Q(300),S(300),QQ(300),X(80),G(80),ZI(10),ZE(10)
      INTEGER NIN(300),NFI(300),IN(15,80),IO(80,15),NP(300)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
C     FUNCTIA DE MINIMIZAT
      FE=0.
      DO 6 I=1,NC
      QQ(I)=Q(I)
      DO 5 JN=1,NI
      DO 1 JN=1,NNI
      N1=IN(JN,JN)
      N2=IN(JN+1,JN)
      IF (N2.EQ.0) GOTO 5
      IF (N1.EQ.NIN(I).AND.N2.EQ.NFI(I)) GOTO 2
      IF (N1.EQ.NFI(I).AND.N2.EQ.NIN(I)) GOTO 3
    1 CONTINUE
    2 K=1
      GOTO 4
    3 K=-1
    4 QQ(I)=QQ(I)+X(JN)*K
    5 CONTINUE
      FE=FE+S(I)/3*ABS(QQ(I))**3
      IF (NP(I).EQ.0) GOTO 6
      FE=FE-(A/3*ABS(QQ(I))**3/NP(I)**2+B/2*ABS(QQ(I))**2
     */NP(I)+C*ABS(QQ(I)))
    6 CONTINUE
C     COMPONENTELE GRADIENTULUI
      DO 9 JN=1,NI
      G(JN)=0.
      DO 7 JN=1,NNI
      IF (IO(JN,JN).EQ.0) GOTO 9
      K=IABS(IO(JN,JN))
      G(JN)=G(JN)+S(K)*IO(JN,JN)/K*ABS(QQ(K))*QQ(K)
      IF (NP(K).EQ.0) GOTO 7
      G(JN)=G(JN)-IO(JN,JN)/K*(A*ABS(QQ(K))**2/NP(K)**2
     *+B*ABS(QQ(K))/NP(K)+C)*QQ(K)/ABS(QQ(K))
    7 CONTINUE
      IF ((NI-JN).GT.NIF) GOTO 9
      DO 8 K=1,NIF
      IF (JN.NE.(NI-NIF+K)) GOTO 8
      G(JN)=G(JN)+ZE(K)-ZI(K)
    8 CONTINUE
    9 CONTINUE
      RETURN
      END

      SUBROUTINE PIERDS (Q,S,H)
C     CALCULUL PIERDERILOR DE SARCINA
      DIMENSION Q(300),S(300),H(300)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      DO 1 I=1,NC
    1 H(I)=S(I)*Q(I)*ABS(Q(I))
      RETURN
      END

      SUBROUTINE VIT (Q,D,V)
C     CALCULUL VITEZELOR DE CURGERE A APEI
      DIMENSION Q(300),D(300),V(300)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,EST,EPS,LIM
      DO 1 I=1,NC
    1 V(I)=1.274*ABS(Q(I))/D(I)**2
      RETURN
      END
```

```
C     ***********************************
C     *PROGRAMUL  OPNELIN*
C     ***********************************
C     DIMENSIONAREA OPTIMALA A RETELELOR INELARE DE DISTRIBUTIE
C          A APEI FOLOSIND MODELUL DE OPTIMIZARE NELINIARA
C
C
C     LISTA NOTATIILOR
C       NZ        -NUMARUL ZONELOR DE CONSUM
C       NC        -NUMARUL TRONSOANELOR DE CONDUCTE
C       NCR       -NUMARUL TRONSOANELOR RETELEI VIRTUAL RAMIFICATE
C       NI        -NUMARUL INELELOR INDEPENDENTE
C       NNI       -NUMARUL MAXIM DE NODURI ALE UNUI INEL
C                  PLUS UN ULTIM NOD FICTIV  'O'
C       NS        -NUMARUL SURSELOR DE ALIMENTARE
C       NIF       -NUMARUL INELELOR FICTIVE CUPRINSE INTRE SURSELE
C                  K SI K+1, UNDE K=1,...,NS-1
C       NNT       -NUMARUL MAXIM DE NODURI ALE TRASEELOR CUPRINSE
C                  INTRE IPP SI NO, EGAL CU NUMARUL NODURILOR
C                  CELUI MAI LUNG TRASEU PLUS UN ULTIM NOD FICTIV 'O'
C       NDS       -NUMARUL DIAMETRELOR STANDARDIZATE
C       A,B,C     -COEFICIENTII FUNCTIEI DE INTERPOLARE A INALTIMII
C                  DE POMPARE A POMPELOR INTEGRATE PE ARTERE
C       DELTA     -RUGOZITATEA ABSOLUTA A CONDUCTELOR, IN MM
C       STEL      -COEFICIENTUL PRELIMINAR DE REZISTENTA
C                  HIDRAULICA AL CONDUCTELOR
C       BETA      -EXPONENTUL DEBITULUI IN RELATIA FUNCTIONALA DE
C                  FORMA GENERALA
C       VMIN      -VITEZA MINIM ADMISIBILA DE CURGERE A APEI, IN M/S
C       ITERM     -NUMARUL MAXIM DE ITERATII
C       EPS       -VALOAREA ADMISA PENTRU REZIDUURI
C       IA,IB,ALFA-PARAMETRII COSTULUI SPECIFIC DE INVESTITIE SAU A
C                  ENERGIEI SPECIFICE INGLOBATE PENTRU CONDUCTE
C       P1,P2     -COTA DE REPARATII SI INTRETINERE PENTRU CONDUCTELE
C                  RETELEI SI RESPECTIV PENTRU STATIILE DE POMPARE
C       F         -COSTUL DE INSTALARE A UNITATII DE PUTERE
C                  ELECTRICA IN STATIILE DE POMPARE, IN LEI/kW
C       SIGMA     -COEFICIENTUL REZERVEI DE PUTERE INSTALATA
C       ETA       -RANDAMENTUL GENERAL AL STATIILOR DE POMPARE
C       TAU       -COEFICIENTUL CE TINE SEAMA DE NUMARUL EFECTIV DE
C                  ORE DE POMPARE T PE PERIOADA UNUI AN (TAU=T/8760)
C       TE        -TARIFUL ENERGIEI ELECTRICE, IN LEI/kWH
C       FI        -SUMA RAPOARTELOR INTRE DEBITUL LUNAR MEDIU SI
C                  DEBITUL POMPAT PE PERIOADA UNUI AN
C       BETAO     -COTA DE AMORTIZARE PENTRU DURATA DE EXPLOATARE TR
C       TC        -DURATA PENTRU CARE SE APLICA CRITERIUL DE
C                  OPTIMIZARE, IN ANI
C       PSI       -PARAMETRUL ECONOMICO-ENERGETIC AL INSTALATIILOR
C                  DE POMPARE
C       NO(J)     -NODURI OBLIGATE, CU COTA PIEZOMETRICA ZO(J),
C                  IN M, CUNOSCUTA
C       NIN(I)    -NODUL INITIAL AL TRONSONULUI I
C       NFI(I)    -NODUL FINAL AL TRONSONULUI I
C       L(I)      -LUNGIMEA TRONSONULUI I, IN M
C       D(I)      -DIAMETRUL TRONSONULUI I, IN M
C       NP(I)     -NUMARUL POMPELOR INTEGRATE PE ARTERA I
C       S(I)      -MODULUL DE REZISTENTA HIDRAULICA AL TRONSONULUI I,
C                  IN S**2/M**5
C       Q(I)      -DEBITUL TRANSPORTAT PE TRONSONUL I, IN MC/S
C       H(I)      -PIERDEREA DE SARCINA PE TRONSONUL I, IN M
C       V(I)      -VITEZA APEI IN TRONSONUL I, IN M/S
C       QC(N)     -DEBITUL DE CONSUM CONCENTRAT, SAU DE ALIMENTARE
C                  IN NODUL N, IN MC/S
C       ZT(N)     -COTA GEODEZICA A NODULUI N, IN M
C       Z(N)      -COTA PIEZOMETRICA IN NODUL N, IN M
C       HD(N)     -PRESIUNEA DISPONIBILA IN NODUL N, IN M
C       ZIPP(K)   -COTA PIEZOMETRICA LA INSTALATIA DE PUNERE SUB
C                        PRESIUNE K, IN M
C       QO(KZ)    -DEBITUL DE ALIMENTARE AL RETELEI, CORESPUNZATOR
C                  ZONEI DE CONSUM KZ, IN MC/S
C       DS(J)     -LISTA DIAMETRELOR STANDARDIZATE, IN M
C       AL(I,KZ)  -LUNGIMEA TRONSONULUI I AFLAT IN ZONA DE CONSUM
C                  KZ, IN M
C       IN(JN,JM) -MATRICEA CUPRINZIND NODURILE JN ALE INELULUI JM
C       ITRAS(KT,IT)-MATRICEA CUPRINZIND NODURILE IT ALE TRASEULUI KT
C
```


Scanned with OKEN Scanner

```
C  ALOCAREA MEMORIEI
      DIMENSION AL(300,5),QO(5),Q(300),D(300),S(300),H(310),
     *V(300),QC(230),ZT(230),HD(230),QD(230),QSP(5),Z(230),
     *ZIPP(10),DS(25),AJ(250,250),T(250),ZO(10)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15),
     *ITRAS(10,70),NO(10)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      EXTERNAL ORIENT,SISLIN,CLAM
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,2) NZ,NC,NCR,NI,NNI,NS,NNT,NDS,DELTA,
     *BETA,STEL,A,B,C,WMIN,ITERM,EPS,IA,IB,ALFA,P1,P2,
     *TR,ETA,F,SIGMA,TAU,F1,TE,TC
    2 FORMAT (815,/,F5.2,F6.3,F8.5,F10.2,2F8.2,/,F5.2,I5,F10.6,
     *2I5,F5.2,/,2F6.3,F5.1,F5.2,F10.1,2F5.2,F6.2,F5.2,F5.1)
      READ (1,4) ((AL(I,KZ),KZ=1,NZ),I=1,NC)
    4 FORMAT (10F6.1)
      READ(1,6) (QO(KZ),KZ=1,NZ)
    6 FORMAT (8F8.3)
      READ (1,8) (QC(N),ZT(N),N=1,NCR)
    8 FORMAT (3(F8.5,F8.2))
      DO 11 I=1,NC
      READ (1,10) NIN(I),NFI(I),NP(I)
   10 FORMAT (3I5)
   11 CONTINUE
      DO 14 JN=1,NI
      READ (1,13) (IN(JN,JM),JM=1,NNI)
   13 FORMAT (15I5)
   14 CONTINUE
      DO 15 KT=1,NS
      READ (1,13) (ITRAS(KT,IT),IT=1,NNT)
   15 CONTINUE
      READ (1,16) (ZIPP(K),K=1,NS)
   16 FORMAT (8F8.2)
      DO 18 J=1,NS
      READ (1,17) NO(J),ZO(J)
   17 FORMAT (I5,F8.3)
   18 CONTINUE
      OPEN (UNIT=1,FILE='diastan.dat')
      READ (1,19) (DS(J),J=1,NDS)
   19 FORMAT (10F6.3)
      CLOSE (UNIT=1)
      WRITE (3,20)
   20 FORMAT (////,22X,36('*'),/,22X,'* P R O G R A M U L
     *'O P N E L I N *',/,22X,36('*'))
      WRITE (3,21)
   21 FORMAT (////,25X,'VERIFICAREA DATELOR DE INTRARE',/,25X,
     *30('*'),///,34X,'DATE GENERALE',/,34X,13('='))
      WRITE (3,22) NZ,NC,NCR,NI,NNI,NS,NNT,NDS,DELTA,
     *BETA,STEL,A,B,C,WMIN,ITERM,EPS,IA,IB,ALFA,P1,P2,
     *TR,ETA,F,SIGMA,TAU,F1,TE,TC
   22 FORMAT (/,7X,'NZ=',I2,2X,'NC=',I3,2X,'NCR=',I3,2X,'NI=',
     *I3,2X,'NNI=',I2,2X,'NS=',I2,//,2X,'NNT=',I2,2X,'NDS=',I2,
     *2X,'DELTA=',F4.2,' MM',2X,'BETA=',F5.3,2X,'STEL=',F7.5,//,
     *7X,'A=',F9.2,2X,'B=',F7.2,2X,'C=',F5.2,2X,'WMIN=',F4.2,
     *' M/S',//,4X,'ITERM=',I3,2X,'EPS=',F8.6,2X,'IA=',I3,2X,
     *'IB=',I4,2X,'ALFA=',F4.2,//,2X,'P1=',F5.3,2X,'P2=',F5.3,2X,
     *'TR=',F4.1,' ANI',2X,'ETA=',F4.2,2X,'F=',F7.1,' LEI/KWH',//,
     *2X,'SIGMA=',F4.2,2X,'TAU=',F4.2,2X,'F1=',F5.2,2X,'TE=',F4.2,
     *' LEI/KWH',2X,'TC=',F5.1)
      WRITE (3,23)
   23 FORMAT (//,21X,'DEBITUL DE ALIMENTARE PE ZONE DE CONSUM',/,
     *21X,39('='),//,32X,'KZ',3X,'QO(KZ), MC/S',/,30X,19('-'))
      DO 26 KZ=1,NZ
      WRITE (3,24) KZ,QO(KZ)
   24 FORMAT (/,30X,I4,3X,F8.5)
      IF (KZ.NE.NZ) GOTO 26
      WRITE (3,25)
   25 FORMAT (30X,19('-'))
   26 CONTINUE
      WRITE (3,27)
   27 FORMAT (//,25X,'LUNGIMEA TRONSOANELOR PE ZONE',/,
     *25X,29('*'))
      WRITE (3,28) (( AL(I,KZ),KZ=1,NZ),I=1,NC)
   28 FORMAT (/,4X,8F8.1)
      WRITE (3,29)
   29 FORMAT (//,31X,'MATRICEA INELELOR',/,31X,17('='))
      DO 31 JM=1,NI
      WRITE (3,30) (IN(JN,JM),JN=1,NNI)
   30 FORMAT (/,5I,12I5)
   31 CONTINUE
      WRITE (3,32)
   32 FORMAT (//,27X,'MATRICEA TRASEELOR IFP-NO',/,27X,25('='))
      DO 34 KT=1,NS
      WRITE (3,33) (ITRAS(KT,IT),IT=1,NNT)
   33 FORMAT (/,4X,14I4)
   34 CONTINUE
      WRITE (3,35)
   35 FORMAT (//,18X,'COTELE PIEZOMETRICE ESTIMATIVE ALE SURSELOR',/,
     *18X,43('='))
      WRITE (3,36) (ZIPP(K),K=1,NS)
   36 FORMAT (/,4X,8F8.2)
      WRITE (3,37)
   37 FORMAT (//,20X,'NODURILE OBLIGATE SI COTELE PIEZOMETRICE'.
     */,19X,39('='),/,34X,'ALE ACESTORA',/,34X,12('='))
      DO 38 J=1,NS
      WRITE (3,110) NO(J),ZO(J)
  110 FORMAT (/,34X,I3,6X,F8.3)
   38 CONTINUE
      WRITE (3,39)
   39 FORMAT (//,28X,'DIAMETRELE STANDARDIZATE',/,28X,24('='))
      WRITE (3,40) (DS(J),J=1,NDS)
   40 FORMAT (/,4X,8F7.3)
C
C   EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,41)
   41 FORMAT (1//.29X,'REZULTATELE CALCULULUI',/,29X.22('*'))
      WRITE (3,120)
  120 FORMAT (//,20X,'REGIMUL DE CURGERE AL APEI IN CONDUCTE:',
     */20X,39('='))
      IF (BETA.NE.2) GO TO 122
      WRITE (3,121)
  121 FORMAT (/,31X,'TURBULENT PATRATIC')
      GO TO 124
  122 WRITE (3,123)
  123 FORMAT (/,29X,'TURBULENT TRANZITORIU')
  124 WRITE (3,125)
  125 FORMAT (//,28X,'CRITERIUL DE OPTIMIZARE:',/,28X,24('='))
      IF (TC.EQ.1.) GO TO 127
      WRITE (3,126)
  126 FORMAT (/,22X,'CHELTUIELI TOTALE ACTUALIZATE MINIME')
      GO TO 131
  127 IF (F.EQ.0) GO TO 129
      WRITE (3,128)
  128 FORMAT (/,28X,'CHELTUIELI ANUALE MINIME')
      GO TO 131
  129 WRITE (3,130)
  130 FORMAT (/,29X,'CONSUM ENERGETIC MINIM')
C   CALCULUL PARAMETRILOR ENERGETICO-ECONOMICI
  131 RK=0,0826*STEL
      BETAO=1/TR
      IF (TC.EQ.1) GOTO 42
      RA=((1+BETAO)**TR-1)/(BETAO*(1+BETAO)**TR)
      GOTO 132
   42 RA=1
  132 ZITA1=RA*P1*TC/TR
      ZITA2=RA*P2*TC/TR
      PSI=9.81/ETA*(F*SIGMA*ZITA2+730*RA*TE*TAU*F1)
      AO=ALFA/5*ZITA1*IB*RK**(ALFA/5)
C   DETERMINAREA REPARTITIEI DEBITELOR DE TRANZIT
      DO 43 I=1,NC
      L(I)=0
      DO 43 KZ=1,NZ
      L(I)=L(I)+AL(I,KZ)
   43 CONTINUE
      CALL REDEC (AL,QO,QC,QSP,QD,Q)
      WRITE (3,44)
   44 FORMAT (//,22X,'DEBITELE SPECIFICE PE ZONE DE CONSUM',
     */,22X,36('='),//,29X,24('-'),/,31X,'KZ',7X,'QSP(KZ), MC/S',
     */,29X,24('-'))
      DO 47 KZ=1,NZ
      WRITE (3,45) KZ,QSP(KZ)
```


Scanned with OKEN Scanner

```
   45 FORMAT (/,29X,I4,6X,F10.6)
      IF (KZ.NE.NZ) GO TO 47
      WRITE (3,46)
   46 FORMAT (29X,24('-'))
   47 CONTINUE
      WRITE (3,48)
   48 FORMAT (//,27X,'CARACTERISTICILE NODURILOR',/,27X,26('='),
     *//,19X,42('-'),/,22X,'N',3X,'ZT(N), M',3X,'QC(N), MC/S',
     *3X,'QD(N), MC/S',/,19X,42('-'))
      DO 51 N=1,NCR
      WRITE (3,49) N,ZT(N),QC(N),QD(N)
   49 FORMAT (/,19X,I3,4X,F6.2,6X,F8.5,7X,F9.5)
      IF (N.NE.NCR) GOTO 51
      WRITE (3,50)
   50 FORMAT (19X,42('-'))
   51 CONTINUE
      WRITE (3,52)
   52 FORMAT (//,25X,'DEBITELE DE TRANZIT OPTIMIZATE',/,25X,30('='),
     *//,16X,49('-'),/,19X,'I',3X,'NIN(I)',2X,'NFI(I)',3X,'NP(I)',2X,
     *'L(I), M',2X,'Q(I), MC/S',/,16X,49('-'))
      DO 55 I=1,NC
      NIF=NS-1
      CALL OPREDEC (IO,NIF,Q,ORIENT)
      WRITE (3,53) I,NIN(I),NFI(I),NP(I),L(I),Q(I)
   53 FORMAT (/,16X,I4,3(4X,I4),4X,F5.1,4X,F8.5)
      IF (I.NE.NC) GOTO 55
      WRITE (3,54)
   54 FORMAT (16X,49('-'))
   55 CONTINUE
C   DETERMINAREA UNEI APROXIMARI INITIALE A PIERDERILOR DE SARCINA
      X=-(ALFA+5.)/5.
      TETA=0.
      DO 56 N=1,NCR
      IF (QC(N).LE.0.) GOTO 56
      TETA=TETA+PSI/AO*QC(N)
   56 CONTINUE
      DO 57 K=1,NS
   57 H(NC+K)=ZIPP(K)
      DO 58 I=1,NC
      SF=L(I)*ABS(Q(I))**(BETA*ALFA/(ALFA+5.))
   58 H(I)=SF*TETA**(1/X)*Q(I)/ABS(Q(I))
      WRITE (3,59)
   59 FORMAT (//,10X,'APROXIMAREA INITIALA A PIERDERILOR '
     *'DE SARCINA',/,10X,46('='))
      WRITE (3,60) (H(I),I=1,NC+NS)
   60 FORMAT (/,6(3X,F8.3))
C   FORMULAREA SI REZOLVAREA SISTEMULUI DE ECUATII NELINIARE
      CALL SINNER (IO,NO,NS,NNT,KOD,A,B,C,BETA,ALFA,X,
     *AO,PSI,QC,Q,ZO,ZIPP,ITRAS,AJ,T,H,SISLIN)
      IF (KOD.EQ.0) GOTO 80
      DO 63 K=1,NS
      I=NC+K
   63 ZIPP(K)=H(I)
C   CALCULUL DIAMETRELOR
      DO 64 I=1,NC
      D(I)=RK**(1/5)*ABS(Q(I))**(BETA/5)/ABS(H(I))**(1/5)
     **L(I)**(1/5)
      CALL DIAS (I,NDS,DS,KK,Q,WMIN,D)
   64 CALL CLAM (I,DELTA,Q,D,BETA,S)
C   CALCULUL CARACTERISTICILOR HIDRAULICE
      CALL ECHIL (IO,NIF,ZIPP,A,B,C,S,DELTA,D,BETA,Q,CLAM)
      CALL PIERDS (Q,S,H)
      CALL VIT (Q,D,V)
      WRITE(3,65)
   65 FORMAT (//,26X,'CARACTERISTICILE HIDRAULICE ',/,26X,28('='),
     *//,8X,61('-'),/,10X,'I',2X,'Q(I), MC/S',2X,'D(I), M',2X,
     *'S(I), S**2/M**5',2X,'H(I), M',2X,'V(I), M/S',/,8X,61('-'))
      DO 68 I=1,NC
      WRITE (3,66) I,Q(I),D(I),S(I),H(I),V(I)
   66 FORMAT (/,8X,I3,3X,F8.5,4X,F5.3,5X,F9.1,5X,F5.2,5X,F7.2)
      IF (I.NE.NC) GOTO 68
      WRITE (3,67)
   67 FORMAT (8X,61('-'))
   68 CONTINUE
      DO 69 K=1,NCR
   69 Z(K)=0.
      NNO=NO(I)
      Z(NNO)=ZO(I)
      HD(NNO)=Z(NNO)-ZT(NNO)
   70 DO 74 I=1,NC
      N1=NIN(I)
      N2=NFI(I)
      KO=1
      IF (Z(N1).NE.0.AND.Z(N2).EQ.0) GOTO 72
      IF (Z(N1).EQ.0.AND.Z(N2).NE.0) GOTO 71
      GOTO 74
   71 N3=N1
      N1=N2
      N2=N3
      KO=-1
   72 Z(N2)=Z(N1)-KO*H(I)
      IF (NP(I).EQ.0) GOTO 73
      HP=A*ABS(Q(I))**2/NP(I)**2+B*ABS(Q(I))/NP(I)+C
      Z(N2)=Z(N2)+KO*HP*Q(I)/ABS(Q(I))
   73 HD(N2)=Z(N2)-ZT(N2)
   74 CONTINUE
      DO 75 I=1,NC
      N1=NIN(I)
      N2=NFI(I)
      IF (Z(N1).NE.0.AND.Z(N2).NE.0) GOTO 75
      GOTO 70
   75 CONTINUE
      WRITE (3,76)
   76 FORMAT(//,27X,'COTELE PIEZOMETRICE NODALE',/,
     *27X,26('='),//,12X,47('-'),/,14X,'I',2X,'NIN',2X,'NFI',
     *2X,'ZIN, M',3X,'ZFI, M',3X,'HDI, M',2X,'HDF, M',/,12X,47('-'))
      DO 79 I=1,NC
      N1=NIN(I)
      N2=NFI(I)
      WRITE (3,77) I,NIN(I),NFI(I),Z(N1),Z(N2),HD(N1),HD(N2)
   77 FORMAT (/,10X,3(2X,I3),2(2X,F7.3),2(2X,F6.3))
      IF (I.NE.NC) GOTO 79
      WRITE (3,78)
   78 FORMAT (12X,47('-'))
   79 CONTINUE
   80 STOP
      END

      SUBROUTINE REDEC (AL,QO,QC,QSP,QD,Q)
C   REPARTITIA DEBITELOR DE CALCUL
      DIMENSION AL(300,5),QO(5),QC(230),QSP(5),Q(300),SAL(300,5),
     *QD(230),QK(230)
      INTEGER NIN(300),NFI(300),IN(15,80),NP(300)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      DO 2 KZ=1,NZ
      SAL(1,KZ)=AL(1,KZ)
      DO 1 I=2,NC
    1 SAL(I,KZ)=SAL(I-1,KZ)+AL(I,KZ)
      QSP(KZ)=QO(KZ)/SAL(NC,KZ)
    2 CONTINUE
      NCR1=NCR+1
      DO 3 I=1,NCR1
      QD(I)=0
      QK(I)=0
    3 CONTINUE
      DO 5 I=1,NC
      DO 4 KZ=1,NZ
      I1=NIN(I)
      I2=NFI(I)
      QD(I1)=QD(I1)+0.5*QSP(KZ)*AL(I,KZ)
      QD(I2)=QD(I2)+0.5*QSP(KZ)*AL(I,KZ)
    4 CONTINUE
    5 CONTINUE
      DO 7 I=1,NC
    7 Q(I)=0
      DO 8 I=1,NCR
      QK(I)=QK(I)+QD(I)-QC(I)
      Q(I)=QK(I)
      I3=NIN(I)
```


Scanned with OKEN Scanner

```
      QK(I3)=QK(I3)+Q(I)
    8 CONTINUE
      RETURN
      END

      SUBROUTINE OPREDEC (IO,NIF,Q,ORIENT)
C   OPTIMIZAREA REPARTITIEI DEBITELOR DE CALCUL
      DIMENSION Q(300),DQ(80)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      CALL ORIENT(IO)
      NII=NI-NIF
      ITER=0
    1 ITER=ITER+1
      DO 4 JM=1,NII
      QQ=0.
      DD=0.
      DO 2 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GO TO 3
      K=IABS(IO(JM,JN))
      DD=DD+L(K)
      QQ=QQ+L(K)*Q(K)*IO(JM,JN)/K
    2 CONTINUE
    3 DQ(JM)=-QQ/DD
    4 CONTINUE
      DO 7 JM=1,NII
      DO 5 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GO TO 6
      K=IABS(IO(JM,JN))
      Q(K)=Q(K)+DQ(JM)*IO(JM,JN)/K
    5 CONTINUE
    6 CONTINUE
    7 CONTINUE
      IF (ITER.GT.ITERM) GO TO 9
      DO 8 JM=1,NII
      IF (ABS(DQ(JM)).GT.0.0001) GO TO 1
    8 CONTINUE
    9 CONTINUE
      RETURN
      END

      SUBROUTINE ORIENT (IO)
C   DETERMINAREA ORIENTARII TRONSOANELOR
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      DO 6 JM=1,NI
      DO 1 JN=1,NNI
    1 IO(JM,JN)=0
      DO 5 JN=1,NNI
      N1=IN(JN,JM)
      N2=IN(JN+1,JM)
      IF (N2.EQ.0) GOTO 6
      DO 2 I=1,NC
      IF (N1.EQ.NIN(I).AND.N2.EQ.NFI(I)) GOTO 3
      IF (N1.EQ.NFI(I).AND.N2.EQ.NIN(I)) GOTO 4
    2 CONTINUE
    3 IO(JM,JN)=1
      GOTO 5
    4 IO(JM,JN)=-1
    5 CONTINUE
    6 CONTINUE
      RETURN
      END

      SUBROUTINE SINNER (IO,NO,NS,NNT,KOD,A,B,C,BETA,ALFA,X,
     *AO,PSI,QC,Q,ZO,ZIPP,ITRAS,AJ,T,H,SISLIN)
C   APLICAREA ALGORITMULUI NEWTON-RAPHSON
      DIMENSION AJ(250,250),T(250),H(310),QC(230),Q(300),ZIPP(10),
     *ZO(10)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15),
     *ITRAS(10,70),NO(10)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      NN=NC+NS
      K=0

C   GENERAREA MATRICEI JACOBIAN SI A VECTORULUI TERMENILOR LIBERI
   50 DO 1 N=1,NN
    1 T(N)=0.
      N=0
      DO 8 KK=1,NCR
      DO 110 J=1,NS
      IF (KK.EQ.NO(J)) GO TO 8
      N=N+1
      IF (QC(N).GT.0) GOTO 5
      DO 3 I=1,NC
      CO=ABS(Q(I))**(BETA*ALFA/5.)*L(I)**(-X)
      IF (N.NE.NFI(I)) GOTO 2
      T(N)=T(N)-CO*H(I)/ABS(H(I))**(1-X)
      GOTO 3
    2 IF (N.NE.NIN(I)) GOTO 3
      T(N)=T(N)+CO*H(I)/ABS(H(I))**(1-X)
    3 CONTINUE
      GOTO 8
    5 T(N)=-PSI/AO*QC(N)
      DO 7 I=1,NC
      CO=ABS(Q(I))**(ALFA*BETA/5)*L(I)**(-X)
      IF (N.NE.NFI(I)) GOTO 6
      T(N)=T(N)-CO*H(I)/ABS(H(I))**(1-X)
      GOTO 7
    6 IF (N.NE.NIN(I)) GOTO 7
      T(N)=T(N)+CO*H(I)/ABS(H(I))**(1-X)
    7 CONTINUE
  110 CONTINUE
    8 CONTINUE
      NIF=NS-1
      DO 12 JM=1,NI
      N=NCR-NS+JM
      DO 10 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 10
      I=IABS(IO(JM,JN))
      T(N)=T(N)-H(I)*IO(JM,JN)/I
      IF (NP(I).EQ.0) GOTO 10
      HP=A*ABS(Q(I))**2/NP(I)**2+B*ABS(Q(I))/NP(I)+C
      IF (IO(JM,JN).LT.0) GOTO 9
      T(N)=T(N)+HP*Q(I)/ABS(Q(I))
      GOTO 10
    9 T(N)=T(N)-HP*Q(I)/ABS(Q(I))
   10 CONTINUE
      IF ((NI-JM).GT.NIF) GOTO 12
      DO 11 K=1,NIF
      IF (JM.NE.(NI-NIF+K)) GOTO 11
      T(N)=T(N)+H(NC+K)-H(NC+K+1)
   11 CONTINUE
   12 CONTINUE
      DO 19 KT=1,NS
      N=NCR-NS+NI+KT
      T(N)=T(N)-H(NC+KT)+ZO(KT)
      DO 18 IT=1,NNT
      DO 17 I=1,NC
      IF (NIN(I).EQ.ITRAS(KT,IT).AND.NFI(I).EQ.ITRAS(KT,IT+1))
     * GOTO 13
      IF (NFI(I).EQ.ITRAS(KT,IT).AND.NIN(I).EQ.ITRAS(KT,IT+1))
     * GOTO 14
      GOTO 17
   13 KO=1
      GOTO 15
   14 KO=-1
   15 T(N)=T(N)+H(I)*KO
      IF (NP(I).EQ.0) GOTO 17
      HP=A*ABS(Q(I))**2/NP(I)**2+B*ABS(Q(I))/NP(I)+C
      IF (KO.LT.0) GOTO 16
      T(N)=T(N)-HP*Q(I)/ABS(Q(I))
      GOTO 17
   16 T(N)=T(N)+HP*Q(I)/ABS(Q(I))
   17 CONTINUE
   18 CONTINUE
   19 CONTINUE
      DO 20 N=1,NN
      DO 20 I=1,NN
   20 AJ(N,I)=0.
      N=0
      DO 24 KK=1,NCR
```


Scanned with OKEN Scanner

```
      DO 120 J=1,NS
      IF (KK.EQ.NO(J)) GO TO 24
      N=N+1
      DO 23 I=1,NC
      IF (N.NE.NIN(I)) GOTO 21
      KO=-1
      GOTO 22
   21 IF (N.NE.NF1(I)) GOTO 23
      KO=1
   22 CO=ABS(Q(I))**(BETA*ALFA/5)*L(I)**(-X)
      AJ(N,I)=X*CO*H(I)/ABS(H(I))**(2-X)*KO
   23 CONTINUE
  120 CONTINUE
   24 CONTINUE
      DO 27 JM=1,NI
      N=NCR-NS+JM
      DO 26 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 26
      I=IABS(IO(JM,JN))
      AJ(N,I)=IO(JM,JN)/I*Q(I)/ABS(Q(I))
      IF ((NI-JM).GT.NIF) GOTO 26
      DO 25 K=1,NIF
      IF (JM.NE.(NI-NIF+K)) GOTO 25
      AJ(N,NC+K)=-1
      AJ(N,NC+K+1)=1
   25 CONTINUE
   26 CONTINUE
   27 CONTINUE
      DO 33 KT=1,NS
      N=NCR-NS+NI+KT
      DO 32 IT=1,NNT
      DO 31 I=1,NC
      IF (NIN(I).EQ.ITRAS(KT,IT).AND.NF1(I).EQ.ITRAS(KT,IT+1))
     * GOTO 28
      IF (NF1(I).EQ.ITRAS(KT,IT).AND.NIN(I).EQ.ITRAS(KT,IT+1))
     * GOTO 29
      GOTO 31
   28 KO=1
      GOTO 30
   29 KO=-1
   30 AJ(N,I)=-KO
   31 CONTINUE
   32 CONTINUE
      AJ(N,NC+KT)=1
   33 CONTINUE
C  DETERMINATEA VECTORULUI DE CORECTIE
      M=0
      DO 34 I=1,NN
      IF (ABS(T(I)).GT.EPS) M=1
   34 CONTINUE
      IF (M.NE.0) GOTO 35
      GOTO 37
   35 CALL SISLIN (NN,KOD,AJ,T)
      IF (KOD.EQ.0) GOTO 41
C  MODIFICAREA VECTORULUI NECUNOSCUTELOR
      DO 36 I=1,NN
   36 H(I)=H(I)+T(I)
      K=K+1
      IF (K.LT.ITERM) GOTO 50
   37 WRITE (3,38) K
   38 FORMAT (//,5X,'NUMARUL DE ITERATII LA REZOLVAREA'
     *' SISTEMULUI NELINIAR =',I4)
      WRITE (3,39)
   39 FORMAT (/,18X,'SOLUTIA SISTEMULUI NELINIAR',/,18X,27('-'))
      WRITE (3,40) (H(I),I=1,NN)
   40 FORMAT (/,5(3X,F10.3))
   41 RETURN
      END

      SUBROUTINE SISLIN (N,KOD,AJ,T)
C  REZOLVAREA SISTEMULUI DE ECUATII LINIARE
      DIMENSION AJ(250,250),T(250)
      K1=N-1
      DO 5 K=1,K1
      AM=0
      DO 1 I=K,N
      IF (ABS(AJ(I,K)).LE.AM) GOTO 1
      AM=ABS(AJ(I,K))
      IO=I
    1 CONTINUE
      IF (AM.LE.0.0000001) GOTO 6
      KOD=1
      DO 2 J=K,N
      X=AJ(K,J)
      AJ(K,J)=AJ(IO,J)
    2 AJ(IO,J)=X
      X=T(K)
      T(K)=T(IO)
      T(IO)=X
      K2=K+1
      DO 4 I=K2,N
      X=AJ(I,K)/AJ(K,K)
      DO 3 J=K,N
    3 AJ(I,J)=AJ(I,J)-AJ(K,J)*X
    4 T(I)=T(I)-T(K)*X
    5 CONTINUE
      IF (AJ(N,N).LT.0.0000001) GOTO 8
      T(N)=T(N)/AJ(N,N)
      K=N-1
    6 K1=K+1
      DO 7 J=K1,N
    7 T(K)=T(K)-AJ(K,J)*T(J)
      T(K)=T(K)/AJ(K,K)
      K=K-1
      IF (K.GE.1) GOTO 6
      GOTO 10
    8 KOD=0
      WRITE (3,9)
    9 FORMAT (/,20X,'SISTEM LINIAR INCOMPATIBIL')
   10 RETURN
      END

      SUBROUTINE DIAS (I,NDS,DS,KK,Q,WMIN,D)
C  ALEGEREA DIAMETRELOR STANDARDIZATE
      DIMENSION DS(30),Q(300),D(300)
      INTEGER KK(300)
      K=1
    1 KK(I)=K
      IF (D(I).LE.DS(K)) GOTO 2
      IF (K.GE.NDS) GOTO 3
      K=K+1
      GOTO 1
    2 D(I)=DS(K)
      W=1.274*ABS(Q(I))/D(I)**2
      IF (W.GE.WMIN) GOTO 4
      K=K-1
      IF (K.EQ.0) GOTO 4
      GOTO 2
    3 D(I)=DS(NDS)
    4 CONTINUE
      RETURN
      END

      SUBROUTINE CLAM (I,DELTA,Q,D,BETA,S)
C  CALCULUL COEFICIENTULUI LAMBDA SI A REZISTENTEI HIDRAULICE S
      DIMENSION Q(300),D(300),S(300)
      INTEGER NIN(300),NF1(300),NP(300),IN(15,80)
      REAL LAMBDA,L(300)
      COMMON NZ,NC,NIN,NF1,L,NI,IN,NCR,NN1,NP,ITERM,EPS
      RALP=1/(-2*ALOG10(DELTA/(1000*D(I)))+1.138)
      IF (BETA.NE.2) GOTO 1
      LAMBDA=RALP**2
      GOTO 2
    1 A2=DELTA*ABS(Q(I))/(1.13*D(I)**2)
      A1=(1274*A2+8)*RALP-4
      LAMBDA=((A1+SQRT((ABS(A1))**2+20380*A2*RALP))/(2550*A2))**2
    2 S(I)=0.0826*LAMBDA*L(I)/D(I)**5
      RETURN
      END

      SUBROUTINE ECHIL (IO,NIF,ZIPP,A,B,C,DELTA,D,BETA,S,Q,CLAM)
C  ECHILIBRAREA DISTRIBUTIEI DEBITELOR PE INELE
      DIMENSION S(300),Q(300),DQ(80),DH(80),SR(80),ZIPP(10),D(300)
```


Scanned with OKEN Scanner

```
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80),IO(80,15),KOD(80)
      REAL L(300),LAMBDA
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      ITER=0
    1 ITER=ITER+1
      DO 6 JM=1,NI
      KOD(JM)=0
      SR(JM)=0
      DH(JM)=0
      DO 3 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 3
      I=IABS(IO(JM,JN))
      SR(JM)=SR(JM)+2*S(I)*ABS(Q(I))
      DH(JM)=DH(JM)+S(I)*Q(I)*ABS(Q(I))*IO(JM,JN)/I
      IF (NP(I).EQ.0) GOTO 3
      SR(JM)=SR(JM)-(2*A*ABS(Q(I))/NP(I)+B)
      HP=A*(ABS(Q(I))**2)/NP(I)**2+B*ABS(Q(I))/NP(I)+C
      IF (IO(JM,JN).LT.0) GOTO 2
      DH(JM)=DH(JM)-HP*Q(I)/ABS(Q(I))
      GOTO 3
    2 DH(JM)=DH(JM)+HP*Q(I)/ABS(Q(I))
    3 CONTINUE
      IF ((NI-JM).GT.NIF) GOTO 5
      DO 4 K=1,NIF
      IF (JM.NE.(NI-NIF+K)) GOTO 4
      DH(JM)=DH(JM)+ZIPP(K+1)-ZIPP(K)
    4 CONTINUE
    5 DQ(JM)=-DH(JM)/SR(JM)
      IF (ABS(DQ(JM)).LE.0.0001) GOTO 7
      DO 6 JN=1,NNI
      IF (IO(JM,JN).EQ.0) GOTO 6
      I=IABS(IO(JM,JN))
      Q(I)=Q(I)+DQ(JM)*IO(JM,JN)/I
      CALL CLAM (I,DELTA,Q,D,BETA,S)
    6 CONTINUE
      GOTO 8
    7 KOD(JM)=1
    8 CONTINUE
      IF (ITER.GE.ITERM) GOTO 10
      DO 9 JM=1,NI
      IF (ABS(KOD(JM)).EQ.0) GOTO 1
    9 CONTINUE
   10 WRITE (3,11) ITER
   11 FORMAT (//,25X,'NUMARUL DE ITERATII LA ECHILIBRARE=',I4)
      RETURN
      END

      SUBROUTINE PIERDS (Q,S,H)
C   CALCULUL PIERDERILOR DE SARCINA
      DIMENSION Q(300),S(300),H(310)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      DO 1 I=1,NC
    1 H(I)=S(I)*Q(I)*ABS(Q(I))
      RETURN
      END

      SUBROUTINE VIT (Q,D,V)
C   CALCULUL VITEZELOR DE CURGERE A APEI
      DIMENSION Q(300),D(300),V(300)
      INTEGER NIN(300),NFI(300),NP(300),IN(15,80)
      REAL L(300)
      COMMON NZ,NC,NIN,NFI,L,NI,IN,NCR,NNI,NP,ITERM,EPS
      DO 1 I=1,NC
    1 V(I)=1.274*ABS(Q(I))/D(I)**2
      RETURN

      END


C     ************************************
C     * P R O G R A M U L  D I R E I N T O *
C     ************************************
C     DIMENSIONAREA OPTIMIZATA A RETELELOR INTERIOARE DE ALIMENTARE
C                        CU APA IN CLADIRI
C
C
C   LISTA NOTATIILOR
C     NC-NUMARUL TRONSOANELOR RETELEI
C     NT-NUMARUL TRASEELOR
C     NNT-NUMARUL MAXIM DE NODURI ALE TRASEELOR,EGAL CU NUMARUL NODURILOR
C         CELUI MAI LUNG TRASEU PLUS UN ULTIM NOD FICTIV '0'
C     DELTA-RUGOZITATEA ABSOLUTA A CONDUCTELOR,IN MM
C     NIU-COEFICIENT DE VISCOZITATE CINEMATICA A APEI,IN MM/S
C     A-COEFICIENT IN FUNCTIE DE REGIMUL DE FURNIZARE AL APEI
C     B-COEFICIENT FUNCTIE DE FELUL APEI (RECE SAU CALDA).
C     C-COEFICIENT FUNCTIE DE DESTINATIA CLADIRII
C     EL-VALOAREA LIMITA A SUMEI ECHIVALENTILOR LA ALEGEREA RELATIEI
C        PENTRU DETERMINAREA DEBITULUI DE CALCUL
C     BETA-RAPORTUL INTRE GROSIMEA SI DIAMETRUL DIAFRAGMELOR UTILIZATE
C     NV-NUMARUL VITEZELOR ECONOMICE (VE) DATE FUNCTIE DE DEBITELE
C        DE CALCUL
C     EPS-VALOAREA LIMITA ADMISA PENTRU DIFERENTA PRESIUNILOR NECESARE
C         IN NODURILE DE ECHILIBRARE,IN M CA
C     NI(I)-NODUL INITIAL AL TRONSONULUI 'I'
C     NF(I)-NODUL FINAL AL TRONSONULUI 'I'
C     L(I)-LUNGIMEA TRONSONULUI 'I',IN M
C     EA(I)-ECHIVALENTUL AFERENT TRONSONULUI 'I'
C     ZITA(I)-COEFICIENTUL DE REZISTENTA LOCALA AL TRONSONULUI 'I'
C     HG(I)-INALTIMEA CORESPUNZATOARE NODULUI INITIAL AL TRONSONULUI 'I',IN M
C     HU(KT)-PRESIUNEA DE UTILIZARE CORESPUNZATOARE TRASEULUI KT,IN M
C     DS(J)-LISTA DIAMETRELOR STANDARDIZATE,IN MM
C     DEB(KV,KQ)-MATRICEA VITEZELOR ECONOMICE,IN M/S SI A DEBITELOR DE
C               CALCUL CORESPUNZATOARE,IN L/S
C     ITRAS(KT,IT)-MATRICEA CUPRINZIND NODURILE IT ALE TRONSOANELOR
C                  TRASEULUI KT
C     E(I)-ECHIVALENTUL DE CALCUL AL TRONSONULUI 'I'
C     Q(I)-DEBITUL DE CALCUL AL TRONSONULUI 'I',IN L/S
C     D(I)-DIAMETRUL TRONSONULUI 'I',IN MM
C     V(I)-VITEZA APEI IN TRONSONUL 'I',IN M/S
C     LAMBDA(I)-COEFICIENTUL DE REZISTENTA LINIARA AL TRONSONULUI 'I'
C     HP(I)-PIERDERI DE SARCINA PE TRONSONUL 'I',IN M
C     HN(KT,N)-PRESIUNEA NECESARA IN NODUL 'N' DE PE TRASEUL KT,IN M
C     DO(KT)-DIAMETRUL DIAFRAGMEI DE PE TRASEUL KT,IN MM
C     QT-DEBITUL TOTAL INTRODUS IN RETEA,IN L/S
C     IA,IB,ALFA-PARAMETRII COSTULUI SPECIFIC AL CONDUCTELOR
C     P1,P2-COTA DE REPARATII SI INTRETINERE PENTRU CONDUCTELE RETELEI
C          SI RESPECTIV PENTRU INSTALATIA DE PUNERE SUB PRESIUNE
C     FI-COTA DE AMORTIZARE
C     ETA-RANDAMENTUL GENERAL AL INSTALATIEI DE POMPARE
C     TAU-COEFICIENTUL GLOBAL DE NEUNIFORMITATE A FUNCTIONARII
C         STATIEI DE RIDICARE A PRESIUNII
C     TE-TARIFUL ENERGIEI ELECTRICE,IN LEI/KWH
C     R-EXPONENTUL DIAMETRULUI IN RELATIA PIERDERII DE SARCINA
C     EPS1-EROAREA DE CALCUL A COEFICIENTULUI LAMBDA
C
C   ALOCAREA MEMORIEI
      DIMENSION EA(350),EK(350),ZITA(350),HG(350),HU(30),DS(15),E(350),
     *Q(350),D(350),V(350),HP(350),SH(350),DO(30),HN(30,50)
      INTEGER NI(350),NF(350),ITRAS(30,50)
      REAL L(350),LAMBDA(350),LAMBDA1,NIU
C
C   INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) NC,NT,NNT,NDS,EPS,DELTA,NIU,A,B,C,EL,BETA,VAD
    1 FORMAT (4I4,F6.3,6F5.2,2F6.3)
      READ (1,110) P1,P2,FI,F,SIGMA,TE,TAU,ETA,R,ALFA,IA,IB,EPS1
  110 FORMAT (3F5.2,F7.1,F4.1,3F6.2,F4.1,F5.2,2I5,F7.5)
      READ (1,2) (DS(J),J=1,NDS)
    2 FORMAT (12F6.1)
      READ (1,3) (HU(KT),KT=1,NT)
    3 FORMAT (14F5,2)
      DO 7 KT=1,NT
      READ (1,6) (ITRAS(KT,IT),IT=1,NNT)
    6 FORMAT (16I4)
    7 CONTINUE
      DO 115 I=1,NC
      READ (1,8) I,NI(I),NF(I),L(I),EA(I),HG(I),ZITA(I)
    8 FORMAT (3I4,3F6.2,F5,1)
  115 CONTINUE
      WRITE (3,9)
    9 FORMAT (////,24X,36('*'),/,24X,'*P R O G R A M U L  D I R E I N T O*',
     */,24X,36('*'))
```


Scanned with OKEN Scanner

```
      WRITE (3,10)
   10 FORMAT (///,22X,'VERIFICAREA PARAMETRILOR DE INTRARE',/,22X,
     *35('='),//,34X,'DATE GENERALE',/,34X,13('*'),/)
      WRITE (3,11) NC,NT,NNT,NDS,EPS,DELTA,NIU,A,B,C,EL,BETA,VAD
   11 FORMAT (/,7X,'NC=',I3,2X,'NT=',I2,2X,'NNT=',I3,2X,'NDS=',I2,2X,
     *'EPS=',F4.2,2X,'DELTA=',F4,2,' MM',2X,'NIU=',F4.2,' MMP/S',2X,'A=',
     *F4.2,//,17X,2X,'B=',F3.1,2X,'C=',F3.1,2X,'EL=',F4.1,2X,'BETA=',
     *F5.3,2X,'VAD=',F4.1,' M/S')
      WRITE (3,120) P1,P2,F1,F,SIGMA,TE TAU,ETA,R,ALFA,IA,IB,EPS1
  120 FORMAT (/,1X,'P1=',F5.2,2X,'P2=',F5,2.2X,'F1=',F5.2,2X,'F=',F7,1,
     *' LEI/KW',2X,'SIGMA=',F4.1,2X,'TE=',F6.2,' LEI/KWH',2X,'TAU=',
     *F6.2,2X,'ETA=',F6.2,//,23 X,'R=',F4.1,2X,'ALFA=',F5.2.2X,'IA=',I5,
     *2X,'IB=',I5,2X,'EPS1=',F7.5)
      WRITE (3,12)
   12 FORMAT (//,26X,'DIAMETRELE STANDARDIZATE,IN MM',/,26X.30('*'),/)
      WRITE (3,13) (DS(J),J=1,NDS)
   13 FORMAT (/,10X,10F6.1)
      WRITE (3,14)
   14 FORMAT (///,18X,'PRESIUNEA DE UTILIZARE PENTRU TRASEELE KT,IN M',
     *,/,18X,45('*'),//,34X,'KT',2X,'HU(KT), M',/,34X,10('-'))
      DO 16 KT=1,NT
      WRITE (3,15) NT,HU(KT)
   15 FORMAT (/,34X,I2,3X.F4.2)
   16 CONTINUE
      WRITE (3,20)
   20 FORMAT (//,32X,'MATRICEA TRASEELOR',/,32X,18('*'),/)
      DO 22 KT=1,NT
      WRITE (3,21) (ITRAS(KT,IT),IT=1,NNT)
   21 FORMAT (/,10X,30I4)
   22 CONTINUE
      WRITE (3,23)
   23 FORMAT (//,25X,'CARACTERISTICILE TRONSOANELOR',/,25X.29('*'),
     *//,17X,'I',5X,'NI(I)',1X,'NF(I)',1X,'L(I), M',1X,'EA(I)',1X,
     *'HG(I), M',2X,'ZITA(I)',/,17X.50('-'))
      DO 25 I=1,NC
      WRITE (3,24) I,NI(I),NF(I),L(I),EA(I),HG(I),ZITA(I)
   24 FORMAT (/,13X,I3,4X,I3,3X,I3,3X,F5.2,4X,F5.2,3X,F5.2,5X,F4.1)
   25 CONTINUE
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,26)
   26 FORMAT (//,29X,'REZULTATELE CALCULULUI',/,29X,22('='),///,17X,
     *'ECHIVALENTII SI DEBITELE DE CALCUL PE TRONSOANE',/,17X,46('-'),
     *//,28X,'I',5X,'E(I)',5X,'Q(I), L/S',/,28X,24('-'))
      DO 30 I=1,NC
      EK(I)=0.
   30 CONTINUE
      DO 31 I=1,NC
      EK(I)=EK(I)+EA(A)
      E(I)=EK(I)'
      I1=NI(I)
      EK(I1)=EK(I1)+E(I)
   31 CONTINUE
      DO 37 I=1,NC
      IF (E(I),GE.EL) GOTO 33
      Q(I)=A*B*E(I)
      GOTO 35
   33 IF (C,GT,1) GOTO 34
      Q(I)=A*B*C*SQRT(E(I))
      GOTO 35
   34 Q(I)=B*(A*C*SQRT(E(I))+0.004*E(I))
   35 WRITE (3,36) I,E(I),Q(I)
   36 FORMAT (/,42X,I3,4X,F5.2,7X,F6.3)
   37 CONTINUE
      QT=Q(NC)
      PSI=9.81/ETA*((P2+F1)*F*SIGMA+8760.*TE*TAU)
      FE=8.*R*PSI/(96.8*ALFA*IB*(P1+F1))
      DO 130 I=1,NC
      LAMBDA1=0.035
   38 D(I)=1000.*(FE*QT/1000.)**(1./(ALFA+R))*LAMBDA1**(1./(ALFA+R))
     **(Q(I)/1000.)**(2./(ALFA+R))
      CALL CLAM (I,DELTA,D,Q,NIU,LAMBDA)
      IF (ABS(LAMBDA(I)-LAMBDA1).LE.EPS1) GOTO 39
      LAMBDA1=LAMBDA(I)
      GOTO 38
   39 J=1
   40 IF (D(I).LE.DS(J)) GOTO 125
      J=J+1
      GOTO 40
  125 D(I)=DS(J)
      CALL CLAM (I,DELTA,D,Q,NIU,LAMBDA)
      CALL VIT (I,Q,D,V)
      CALL PIERDS (I,LAMBDA,L,D,ZITA,Q,HP)
  130 CONTINUE
      WRITE (3,135)
  135 FORMAT (///,8X,'DIAMETRELE, VITEZELE SI PIERDERILE DE '
     *' SARCINA PE TRONSOANE',/,8X,57('-'))
      WRITE (3,140)
  140 FORMAT (///,2X,'I',5X,'L(I), M',3X,'Q(I), L/S',3X,'D(I), MM',
     *3X,'V(I), M/S',3X,'LAMBDA(I)',3X,'HP(I), M',3X,'SH(N), M',
     *,/,2X,82('-'))
      DO 41 KT=1,NT
      DO(KT)=0.
   41 CONTINUE
      DO 42 KT=1,NT
      CALL SUMPER (NNT,KT,ITRAS,HG,HU,HP,SH,HN)
   42 CONTINUE
      KT=1
   43 IT=1
   44 N3=ITRAS(KT,IT)
      IF (N3.EQ.0) GOTO 48
      KTT=KT+1
   45 ITT=2
   46 N4=ITRAS(KTT,ITT)
      N5=ITRAS(KTT,ITT-1)
      IF (N4.EQ.0) GOTO 49
      IF (N3.EQ.N4) GOTO 51
   47 ITT=ITT+1
      GOTO 46
   48 IF (KT.EQ.(NT-1)) GOTO 59
      KT=KT+1
      GOTO 43
   49 IF (KTT.EQ.NT) GOTO 50
      KTT=KTT+1
      GOTO 45
   50 IT=IT+1
      GOTO 44
   51 IF (ABS(HN(KT,N3)-HN(KTT,N4)).LE.EPS) GOTO 47
      ITT=1
   52 I=ITRAS(KTT,ITT)
      J=1
   53 IF (DS(J).EQ.D(I)) GOTO 54
      J=J+1
      GOTO 53
   54 IF (J.EQ.1) GOTO 55
      D(I)=DS(J-1)
      CALL CLAM (I,DELTA,D,Q,NIU,LAMBDA)
      CALL PIERDS (I,LAMBDA,L,D,ZITA,Q,HP)
      CALL VIT (I,Q,D,V)
      IF (V(I).LE.VAD) GOTO 56
      D(I)=DS(J)
   55 ITT=ITT+1
      I1=ITRAS(KTT,ITT+1)
      IF (I1.EQ.0) GOTO 57
      GOTO 52
   56 CALL SUMPER (NNT,KTT,ITRAS,HG,HU,HP,SH,HN)
      DH=HN(KT,N3)-HN(KTT,N4)
      IF (ABS(DH).LE.EPS) GOTO 47
      IF (DH.GT.0) GOTO 55
      D(I)=DS(J)
      CALL CLAM (I,DELTA,D,Q,NIU,LAMBDA)
      CALL PIERDS (I,LAMBDA,L,D,ZITA,Q,HP)
      CALL VIT (I,Q,D,V)
      CALL SUMPER (NNT,KTT,ITRAS,HG,HU,HP,SH,HN)
      DH=HN(KT,N3)-HN(KTT,N4)
   57 ALFA=10.5-1.3*BETA
      DO(KTT)=ALFA*2.58*SQRT(Q(N5)/DH**0.5)
      GOTO 49
   59 DO 61 I=1,NC
      WRITE (3,60) I,L(I),Q(I),D(I),V(I),LAMBDA(I),HP(I),SH(I)
   60 FORMAT (/,1X,I2,6X,F6.2,6X,F5.2,6X,F5.2,6X,F4.1,6X,F6.4,5X,F6.3,
     *5X,F6.3)
   61 CONTINUE
      WRITE (3,64)
```


Scanned with OKEN Scanner

```
   64 FORMAT (//,26X,'PRESIUNILE NECESARE IN NODURI',/,26X,29('-'),
     *//,28X,'KT',5X,'N',5X,'HN(KT,N), M',/,28X,25('-'))
      DO 67 KT=1,NT
      DO 66 IT=1,NNT
      N1=ITRAS(KT,IT)
      N2=ITRAS(KT,IT+1)
      IF (N2.EQ.0) GOTO 67
      WRITE (3,65) KT,N2,HN(KT,N2)
   65 FORMAT (/,44X,I2,4X,I3,8X,F7.3)
   66 CONTINUE
   67 CONTINUE
      WRITE (3,71)
   71 FORMAT (///,24X,'DIAMETRELE DIAFRAGMELOR PE TRASEE',/,24X,33('-'),
     *//,34X,'KT',3X,'DO(KT), MM',/,34X,14('-'))
      DO 73 KT=1,NT
      WRITE (3,72) KT,DO(KT)
   72 FORMAT (/,34X,I2,5X,F6.2)
   73 CONTINUE
      HNEC=HN(1,NC+1)
      WRITE (3,75) HNEC
   75 FORMAT (//,31X,'PRESIUNEA NECESARA',/,31X,18('*'),//,35X,
     *'HNEC=',F5.2,' M')
      CC=0.
      DO 80 I=1,NC
      CC=CC+((A+(B*(D(I)/1000.))**ALFA*L(I)
   80 CONTINUE
      CP=9.81/ETA*F*SIGMA*HN(1,NC+1)*(QT/1000.)
      CE=86000./ETA*TE*TAU*HN(1,NC+1)*(QT/1000.)
      CA=(P1+F1)*CC+(P2+F1)*CP+CE
      WRITE (3,90) CC,CE,CA
   90 FORMAT (///,19X,'COSTUL DE INVESTITIE AL CONDUCTELOR CC=',F10.1,
     *' LEI',/,19X,35('*'),//,19X,'COSTUL ANUAL AL ENERGIEI ELECTRICE
     * CE=',F10.1,' LEI/AN',/,19X,34('*'),//,19X,'CHELTUIELILE ANUALE
     * TOTALE  CA=',F10.1,' LEI/AN',/,19X,26('*'))
      STOP
      END

      SUBROUTINE PIERDS (I,LAMBDA,L,D,ZITA,Q,HP)
C     CALCULUL PIERDERILOR DE SARCINA PE TRONSOANE
      DIMENSION D(350),ZITA(350),Q(350),HP(350)
      REAL LAMBDA(350),L(350)
      A2=((LAMBDA(I)*1000.*L(I))/D(I))+ZITA(I))
      HP(I)=((82627.*Q(I)**2.)/D(I)**4.)*A2
      RETURN
      END



      SUBROUTINE CLAM (I,DELTA,D,Q,NIU,LAMBDA)
C     CALCULUL COEFICIENTULUI DE REZISTENTA LINIARA
      DIMENSION D(350),Q(350)
      REAL NIU,LAMBDA(350)
      RALP=1./(-2.*ALOG10(DELTA/D(I))+1.138)
      AO=DELTA*(Q(I)/1000.)/(NIU*(D(I)/1000.)**2.)
      A1=(1274.*AO+8.)*RALP-4
      LAMBDA(I)=((A1+SQRT(A1**2.+20380.*AO*RALP))/(2550.*AO))**2.
      RETURN
      END



      SUBROUTINE VIT (I,Q,D,V)
C     CALCULUL VITEZELOR APEI IN TRONSOANE
      DIMENSION Q(350),D(350),V(350)
      V(I)=1274.*Q(I)/D(I)**2.
      RETURN
      END



      SUBROUTINE SUMPER (NNT,KT,ITRAS,HG,HU,HP,SH,HN)
C     DETERMINAREA SUMEI PIERDERILOR DE SARCINA
      DIMENSION HG(350),HU(350),HP(350),SH(350),ITRAS(30,50),HN(30,50)
      DO 3 IT=1,NNT
      N1=ITRAS(KT,IT)
      N2=ITRAS(KT,IT+1)
      IF (N2.EQ.0) GOTO 4
      IF (IT.EQ.1) GOTO 1
      GOTO 2
    1 SH(N1)=HU(KT)
      HN(KT,N1)=SH(N1)
    2 SH(N2)=SH(N1)+HP(N1)
      HN(KT,N2)=SH(N2)+HG(N1)
    3 CONTINUE
    4 CONTINUE
      RETURN
      END



C     ************************************
C     * P R O G R A M U L  D E F R I Z O P *
C     ************************************
C     DIMENSIONAREA OPTIMALA A INCAPERILOR FRIGORIFICE
C
C
C   LISTA NOTATIILOR
C     NI-NUMARUL INCAPERILOR FRIGORIFICE
C     NS-NUMARUL STRATURILOR DE MATERIAL CARE ALCATUIESC UN ELEMENT
C     BO,B-COEFICIENTI DE VARIATIE A CONDUCTIVITATII TERMICE A IZOLATIEI,
C          LAMBDAIZ,CU TEMPERATURA MEDIE A IZOLATIEI
C     CIZ-COSTUL SPECIFIC AL IZOLATIEI,IN LEI/MC
C     EIZ-ENERGIA SPECIFICA INGLOBATA IN IZOLATIE,IN KWH/MC
C     T-DURATA NORMATA DE RECUPERARE A IZOLATIEI,IN ANI
C     TAU-NUMARUL DE ORE NECESARE FURNIZARII FRIGULUI PE PERIOADA UNUI AN.
C        IN ORE
C     P-COTA DE AMORTIZARE REPARATII SI INTRETINERE PENTRU IZOLATIE
C     AO-COEFICIENTUL DE ACTUALIZARE AL CHELTUIELILOR ANUALE
C     SIGMA-COEFICIENTUL PIERDERILOR SUPLIMENTARE DE FRIG DATORITA
C          REGIMULUI NESTATIONAR
C     CF-PRETUL DE COST AL ENERGIEI FRIGORIFICE,IN LEI/KWH
C     WI-VITEZA AERULUI INTERIOR UNUI PERETE,IN M/S
C     WE-VITEZA AERULUI EXTERIOR UNUI PERETE,IN M/S
C     LAMBDAI-CONDUCTIVITATEA TERMICA A AERULUI IN INTERIORUL SPATIULUI
C           RACIT,IN W/(M.GRD)
C     LAMBDAE-CONDUCTIVITATEA TERMICA A AERULUI EXTERIOR,INW/(M.GRD)
C     NIUI-VISCOZITATEA CINEMATICA A AERULUI INTERIOR,IN MMP/S
C     NIUE-VISCOZITATEA CINEMATICA A AERULUI EXTERIOR,IN MMP/S
C     AI-DIFUZIVITATEA AERULUI INTERIOR,IN MMP/S
C     AE-DIFUZIVITATEA AERULUI EXTERIOR,IN MMP/S
C     TPI-TEMPERATURA SUPRAFETEI INTERIOARE A UNUI PERETE,IN GRD
C     EPS-PRECIZIA DE CALCUL A TEMPERATURII UNUI PERETE,IN GRD
C     TC,TV-TEMPERATURA DE CONDENSARE SI DE VAPORIZARE A AGENTULUI
C          FRIGORIFIC,IN GRD
C     ETAE-RANDAMENTUL MEDIU DE OBTINERE A ENERGIEI ELECTRICE DIN
C          ENERGIA PRIMARA
C     ETAF-RADAMENTUL CICLULUI FRIGORIFIC REAL FATA DE UN CICLU CARNOT
C          PARCURS INTRE ACELEASI TEMPERATURI
C     PSI-COEFICIENT DE PONDERE CRITERIALA
C     TZ-TEMPERATURA MEDIE ZILNICA A AERULUI EXTERIOR,IN GRD
C     AZ-COEFICIENTUL DE AMORTIZARE A OSCILATIEI ZILNICE A TEMPERATURII
C        AERULUI EXTERIOR,IN GRD
C     TS-TEMPERATURA SOLULUI,IN GRD
C     DELTA(M)-GROSIMEA STRATULUI M DE MATERIAL AL UNUI PERETE,IN M
C     LAMBDA(M)-CONDUCTIVITATEA TERMICA A STRATULUI M DE MATERIAL,
C              IN W/(M.GRD)
C     H-INALTIMEA SAU LATIMEA ELEMENTULUI,IN M
C     L-LUNGIMEA ELEMENTULUI,IN M
C     TI(I)-TEMPERATURA SPATIULUI RACIT,IN GRD
C     NE(I)-NUMARUL ELEMENTELOR DE CONSTRUCTIE ALE INCAPERII I
C     TE(I,J)-TEMPERATURA EXTERIOARA DE CALCUL PENTRU PERETELE J AL
C            INCAPERII I,IN GRD
C     ALFAI(I,J)-COEFICIENTUL SCHIMBULUI DE CALDURA DIN INTERIORUL
C               SPATIULUI RACIT,IN W/(MP.GRD)
C     ALFAE(I,J)-COEFICIENTUL SCHIMBULUI DE CALDURA DIN EXTERIOR,
C               IN W/(MP.GRD)
C     K(I,J)-COEFICIENTUL DE TRANSMITERE A CALDURII PRIN PERETELE J AL
C           INCAPERII I,IN W/(MP,GRD)
C     DELTAIZ(I,J)-GROSIMEA IZOLATIEI PERETELUI J AL INCAPERII I,IN MM
C     KOD-IDENTIFICATOR CU VALOAREA O SAU 1 DUPA CUM SE FOLOSESTE LA
C        CALCULUL IZOLATIEI METODA OPTIMIZARII SAU METODA CLASICA
C
C   ALOCAREA MEMORIEI
      DIMENSION TI(50),DELTA(20),TE(50,8),ALFAI(50,8),ALFAE(50,8),
```


Scanned with OKEN Scanner

```
     *DELTAIZ(50,8),TT(50)
      INTEGER ITI(50,8),ITE(50,8),NE(50)
      REAL NIUI,NIUE,LAMBDAI,LAMBDAE,LAMBDAIZ,L,LAMBDA(20),K(50,8)
C
C     INTRODUCEREA DATELOR SI SCRIEREA LOR
      READ (1,1) NI,TZ,TS,AZ,BO,B,EPS,KOD
    1 FORMAT (I5,3F5.1,F10.3,F10.5,F5.1,I5)
      READ (1,96) (NE(I),I=1,NI)
   96 FORMAT (26I3)
      IF (KOD.EQ.0.) GOTO 3
      DO 97 I=1,NI
      NN=NE(I)
      READ (1,2) (K(I,J),J=1,NN)
    2 FORMAT (16F5.2)
   97 CONTINUE
      GOTO 5
    3 READ (1,4) CIZ,EIZ,P,T,AO,SIGMA,TAU,CF,ETAE,ETAF,PSI
    4 FORMAT (2F10.1,F5.2,F5.1,2F5.2,F10.1,4F5.2)
    5 WRITE (3,6)
    6 FORMAT (////,22X,36('*'),/,22X,'*P R O G R A M U L  D E F R I Z O P*,
     */,22X,36('*'))
      WRITE (3,90)
   90 FORMAT (//,12X,'CALCULUL IZOLATIEI INCAPERILOR FRIGORIFICE',/,12X,
     *42('-'))
      WRITE (3,7)
    7 FORMAT (///,17X,'VERIFICAREA PARAMETRILOR DE INTRARE',/,17X,35('='),
     *///,27X,'DATE GENERALE',/,27X,13('*'))
      WRITE (3,8) NI,TZ,TS,AZ,BO,B,EPS,KOD
    8 FORMAT (/,11X,'NI=',I3,2X,'TZ=',F5.1,' GRD',2X,'TS=',F5.1,' GRD',
     *2X,'AZ=',F4.1,' GRD',//,13X,'BO=',F5.3,2X,'B=',F7.5,2X,'EPS=',
     *F3.1,2X,'KOD=',I1)
      IF (KOD.EQ.0) GOTO 10
      WRITE (3,9)
    9 FORMAT (///,15X,'COEFICIENTII DE TRANSMITERE A CALDURII',/,15X,
     *38('*'),//,20X,27('-'),/,20X,'I',1X,'I',2X,'J',2X,
     *'K(I,J), W/(MP.GRD)','I',/,20X,'I',25('-'),'I')
      DO 47 I=1,NI
      NN=NE(I)
      DO 46 J=1,NN
      WRITE (3,45) I,J,K(I,J)
   45 FORMAT (20X,'I',2X,'I',/,20X,'I',I2,2X,I1,5X,F8.3,7X,'I')
      IF (J.NE.NN) GOTO 46
      WRITE (3,49)
   49 FORMAT (20X,27('-'))
   46 CONTINUE
   47 CONTINUE
      GOTO 13
   10 WRITE (3,11)
   11 FORMAT (///,19X,'PARAMETRII ECONOMICO-ENERGETICI',/,19X,31('*'
      WRITE (3,12) CIZ,EIZ,P,T,AO,SIGMA,TAU,CF,ETAE,ETAF,PSI
   12 FORMAT (/,5X,'CIZ=',F7.1,' LEI/MC',2X,'EIZ=',F6.1,' KWH/MC',2X,
     *'P=',F4.2,2X,'T=',F4.1,' ANI',2X,'AO=',F4.2,//,1X,'SIGMA=',F4.2,
     *2X,'TAU=',F6.1,' ORE/AN',2X,'CF=',F4.2,' LEI/KWH',2X,'ETAE=',F4.2,
     *2X,'ETAF=',F4.2;2X,'PSI=',F4.2)
   13 WRITE (3,14)
   14 FORMAT (///,11X,'DATE CARACTERISTICE INCAPERILOR SI PERETILOR',
     */,11X,44('*'))
      IF (KOD.NE.0.) GOTO 84
      TC=TZ+AZ+5.
   84 DO 80 I=1,NI
      NN=NE(I)
      DO 79 J=1,NN
      ITI(I,J)=0.
      ITE(I,J)=0.
   79 CONTINUE
   80 CONTINUE
      DO 85 I=1,NI
      READ (1,15) WI,TI(I)
   15 FORMAT (F5.1,F10.1)
      IF (KOD.NE.0) GOTO 81
      TV=TI(I)-5.
   81 CALL PARAM (I,TI,LAMBDAI,NIUI,AI)
      WRITE (3,82) I,WI,TI(I)
   82 FORMAT (//,28X,'INCAPEREA',I3,/,28X,12('='),/,21X,'WI=',F4.1,' M/S
     *3X,'TI=',F6.1,' GRD')
      NN=NE(I)
      DO 40 J=1,NN
      READ (1,83) TE(I,J),NS,WE,H,L
   83 FORMAT (F5.1,I5,F5.1,2F10.2)
      READ (1,16) (DELTA(M),LAMBDA(M),M=1,NS)
   16 FORMAYT (10F8.4)
      IF (J.EQ.NN) GOTO 73
      IF (J.EQ.(NN-1)) GOTO 71
      WRITE (3,70) J
   70 FORMAT (//,28X,'PERETELE',I2,/,28X,10('-'))
      GOTO 76
   71 WRITE (3,72) J
   72 FORMAT (//,28X,'PLAFONUL',I2,/,28X,10('-'))
      GOTO 76
   73 WRITE (3,92) J
   92 FORMAT (//,28X,'PARDOSEA',I2,/,28X,10('-'))
   76 WRITE (3,17) NS,WE,H,L,TE(I,J)
   17 FORMAT (/,7X,'NS=',I2,2X,'WE=',F4.1,' M/S',2X,'H=',F4.2,,' M',2X,
     *'L=',F5.2,' M',2X,'TE=',F5.1,' GRD')
      WRITE (3,18)
   18 FORMAT (//,13X,41('-'),/,13X,'I',1X,'M',3X,'DELTA(M), M',3X,
     *'LAMBDA(M), W/(M.GRD)','I',/,13X,'I',39('-'),'I')
      DO 20 M=1,NS
      WRITE (3,19) M,DELTA(M),LAMBDA(M)
   19 FORMAT (13X,'I',3X,'I',/,13X,'I',I2,6X,F5.3,12X,F6.3,8X,'I')
      IF (M.NE.NS) GOTO 20
      WRITE (3,22)
   22 FORMAT (13X,41('-'))
   20 CONTINUE
C
C     EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      IF (TE(I,J).EQ.(TZ+AZ)) GOTO 89
      IF (TE(I,J).EQ.TS) GOTO 89
      TT(I)=TE(I,J)
      CALL PARAM (I,TT,LAMBDAE,NIUE,AE)
   89 LAMBDAIZ=BO+B*(TI(I)+TE(I,J))/2.
      S=0
      DO 21 M=1,NS
      S=S+DELTA(M)/LAMBDA(M)
   21 CONTINUE
      TPI=TI(I)+10.
   30 IF (WI.EQ.0.) GOTO 50
      CALL CONVECTF (I,J,H,L,WI,AI.NIUI,LAMBDAI,ALFAI)
      GOTO 60
   50 CALL CONVECTL (I,J,H,TI,TPI,AI,NIUI,LAMBDAI,DELTATI,ALFAI)
   60 IF (TE(I,J).NE.(TZ+AZ)) GOTO 88
      ALFAE(I,J)=17.5
      GOTO 24
   88 IF (TE(I,J).NE.TS) GOTO 74
      ALFAE(I,J)=9999.9
      GOTO 24
   74 TPE=TE(I,J)-11.
   28 IF (WE.NE.0.) GOTO 75
      TT(I)=TE(I,J)
      CALL CONVECTL (I,J,H,TT,TPE,AE,NIUE,LAMBDAE,DELTATE,ALFAE)
      GOTO 24
   75 CALL CONVECTF (I,J,H,L,WE,AE,NIUE.LAMBDAE,ALFAE)
   24 IF (KOD.EQ.0.) GOTO 25
      DELTAIZ(I,J)=LAMBDAIZ/K(I,J)-(1./ALFAI(I,J)+1./ALFAE(I,J)+S)*LAMBDAIZ
      D5=1./ALFAI(I,J)+1./ALFAE(I,J)+S
      GOTO 29
   25 A=((1.+AO)**T-1.)/(AO*(1.+AO)**T)
      D1=PSI*A*SIGMA*CF
      D2=T*(1.-PSI)*(TC-TV)/(ETAE*ETAF*(TV+273))
      D3=(TE(I,J)-TI(I))*TAU*LAMBDAIZ/1000.
      D4=PSI*(A*P+1.)*CIZ+(1.-PSI)*EIZ
      D5=1./ALFAI(I,J)+1./ALFAE(I,J)+S
      DELTAIZ(I,J)=SQRT((D1+D2)*D3/D4)-LAMBDAIZ*D5
   29 K(I,J)=1./(D5+DELTAIZ(I,J)/LAMBDAIZ)
      IF (WI.NE.0.) GOTO 31
      YI=K(I,J)*(TE(I,J)-TI(I))/ALFAI(I,J)
      IF (ABS(DELTATI-YI).LE.EPS) GOTO 31
      TPI=TPI-EPS
      ITI(I,J)=ITI(I,J)+1
      GOTO 30
   31 IF (TE(I,J).EQ.(TZ+AZ)) GOTO 32
      IF (TE(I,J).EQ.TS) GOTO 32
      IF (WE.NE.0.) GOTO 32
```


Scanned with OKEN Scanner

```
      YE=K(I,J)*(TI(I)-TE(I,J))/ALFAE(I,J)
      IF (ABS(DELTATE-YE).LE.EPS) GOTO 32
      TPE=TPE+EPS
      ITE(I,J)=ITE(I,J)+1
      GOTO 28
   32 DELTAIZ(I,J)=DELTAIZ(I,J)*1000.
      IF (KOD.EQ.0) GOTO 40
      DELTAIZ(I,J)=DELTAIZ(I,J)*1.1
   40 CONTINUE
   85 CONTINUE
      WRITE (3,33)
   33 FORMAT (///,29X,'REZULTATELE CALCULULUI',/,29X,22('='))
      WRITE (3,34)
   34 FORMAT (//,1X,86('_'),/,1X,'I',1X,'I',2X,'J',2X,'TI(I)',2X,
     *'TE(I,J)',2X,'ALFAI(I,J)',2X,'ALFAE(I,J)',2X,'K(I,J)',2X,
     *'DELTAIZ(I,J)',2X,'ITI',2X,'ITE',1X,'I',/,1X,'I',8X,'GRD',6X,'GRD',
     *5X,'W/(MP.GRD)',3X,'W/(MP.GRD)',2X,'W/(M.GRD)',6X,'MM',9X,'-',4X,
     *'-',2X,'I',/,1X,'I',84('-'),'I')
      DO 86 I=1,NI
      NN=NE(I)
      DO 36 J=1,NN
      WRITE (3,35) I,J,TI(I),TE(I,J),ALFAI(I,J),ALFAE(I,J),K(I,J),
     *DELTAIZ(I,J),ITI(I,J),ITE(I,J)
   35 FORMAT (1X,'I',84X,'I',/,1X,'I',I2,2X,I1,2X,F5.1,5X,F4.1,7X,F5.2,6X,
     *F7.2,5X,F6.3,7X,F5.1,6X,I3,2X,I3,1X,'I')
      IF (J.NE.NN) GOTO 36
      WRITE (3,87)
   87 FORMAT (1X,86('-'),///)
   36 CONTINUE
   86 CONTINUE
  100 STOP
      END



      SUBROUTINE PARAM (I,TT,LAMB,NIU,A)
C     CALCULUL PARAMETRILOR TERMOFIZICI AI AERULUI
      DIMENSION TT(50)
      REAL LAMB,NIU
      LAMB=0.000076*TT(I)+0.0244
      NIU=0.0834*TT(I)+13.28
      A=0.136*TT(I)+18.9
      RETURN
      END



      SUBROUTINE CONVECL (I,J,H,TT,TP,A,NIU,LAMB,DELTAT,ALFA)
C     CALCULUL COEFICIENTULUI DE CONVECTIE LIBERA
      DIMENSION TT(50),ALFA(50,8)
      REAL NIU,LAMB
      DELTAT=TP-TT(I)
      BETA=1./((TT(I)+TP)/2.+273.)
      PRGR=9.81*BETA*ABS(DELTAT)*H**3./(A*NIU)*1.E+12
      IF (PRGR.GT.0.001) GOTO 1
      C=0.5
      R=0.
      GOTO 4
    1 IF (PRGR.GT.500) GOTO 2
      C=1.18
      R=0.125
      GOTO 4
      IF (PRGR.GT.(2.*10**7.)) GOTO 3
      C=0.54
      R=0.25
      GOTO 4
    3 C=0.135
      R=0.33
    4 ALFA(I,J)=LAMB/H*C*PRGR**R
      RETURN
      END



      SUBROUTINE CONVECTF (I,J,H,L,K,A,NIU,LAMB,ALFA)
C     CALCULUL COEFICIENTULUI DE CONVECTIE FORTATA
      DIMENSION ALFA (50,8)
      REAL NIU,LAMB,L
      DEC=2,*H*L/(H+L)
      PR=NIU/A
      RE1=W*DEC*10**6./NIU
      ALFA1=LAMB/DEC*0.036*RE1**0.8*PR**0,75
      RE2=W*L*10**6./NIU
      ALFA2=LAMB/L*0.036*RE2**0.8*PR**0.75
      IF (ALFA2.GE.ALFA1) GOTO 1
      ALFA(I,J)=ALFA1
      GOTO 2
    1 ALFA(I,J)=ALFA2
    2 CONTINUE
      RETURN
      END



C     ************************************
C     * P R O G R A M U L  C O F R I Z O P *
C     ************************************
C     CALCULUL HIDRAULIC AL CONDUCTELOR FRIGORIFICE SI DIMENSIONAREA
C                 OPTIMALA A IZOLATIEI ACESTORA
C
C
C   LISTA NOTATIILOR
C     NC-NUMARUL CONDUCTELOR
C     T-DURATA NORMATA DE RECUPERARE A IZOLATIEI,IN ANI
C     TAU-NUMARUL ORELOR DE FUNCTIONARE A CONDUCTELOR PE UN AN,ORE/AN
C     SIGMA-COEFICIENTUL APORTURILOR SUPLIMENTARE DE CALDURA PRIN
C           ARMATURILE CONDUCTEI
C     P-COTA DE AMORTIZARE,REPARATIE SI INTRETINERE PENTRU IZOLATIE
C     AO-COEFICIENT DE ACTUALIZARE A CHELTUIELILOR ANUALE
C     CF-COSTUL ENERGIEI FRIGORIFICE,IN LEI/KWH
C     LAMC-CONDUCTIVITATEA TERMICA A CONDUCTEI,IN W/(M,GRD)
C     DELP-GROSIMEA STRATULUI PROTECTOR AL IZOLATIEI,IN M
C     LAMP-CONDUCTIVITATEA TERMICA A STRATULUI PROTECTOR,IN W/(M,GRD)
C     BO,B-COEFICIENTI DE VARIATIE A CONDUCTIVITATII TERMICE A
C          IZOLATIEI CU TEMPERATURA
C     LAMIZ-CONDUCTIVITATEA TERMICA A IZOLATIEI,IN W/(M,GRD)
C     EIZ-ENERGIA SPECIFICA INGLOBATA IN IZOLATIE,IN KWH/MC
C     CIZ-INVESTITIA SPECIFICA IN IZOLATIE,IN LEI/MC
C     EPS-MODULUL DE VARITIE A GROSIMII IZOLATIEI,IN M
C     PSI-COEFICIENT DE PONDERE CRITERIALA
C     DE-DIAMETRUL EXTERIOR AL CONDUCTEI,IN M
C     ALFAI-COEFICIENTUL SCHIMBULUI DE CALDURA IN INTERIORUL CONDUCTEI
C           IN W/(MP.GRD)
C     ALFAE-COEFICIENTUL SCHIMBULUI DE CALDURA DIN EXTERIOR,INW/(MP.GRD)
C     W(J)-VITEZA VINTULUI PENTRU TRONSONUL J,IN M/S
C     TI(J)-TEMPERATURA AGENTUUI FRIGORIFIC,IN GRADE
C     TE(J)-TEMPERATURA EXTERIOARA DE CALCUL,IN GRADE
C     TPR(J)-TEMPERATURA PUNCTULUI DE ROUA.IN GRADE,CARE IN CAZUL APLICARII
C           METODEI CLASICE,CIND TEMPERATURA AGENTULUI ESTE MAI MARE SAU
C           EGALA CU TEMPERATURA EXTERIOARA,PRIMESTE SEMNIFICATIA DE
C           REZISTENTA TERMICA TOTALA ADMISA LA TRANSFERUL CALDURII,
C           IN M.GRD/W
C     LAMBDA(J)-CONDUCTIVITATEA TERMICA A AGENTULUI,IN W/(M.GRD)
C     NIU(J)-VISCOZITATEA CINEMATICA A AGENTULUI,IN MMP/S
C     A(J)-DIFUZVITATEA TERMICA A AGENTULU,IN MMP/S
C     L(J)-LUNGIMEA TRONSONULUI J,IN M
C     G(J)-DEBITUL DE AGENT FRIGORIFIC DIN TRONSONUL J,IN MC/S
C     RO(J)-DENSITATEA AGENTULUI FRIGORIFIC DIN TRONSONUL J,IN KG/MC
C     VI(J)-VITEZA INITIALA A AGENTULUI FRIGORIFIC DIN TRONSONUL J,IN M/S
C     ZITA(J)-COEFICIENTUL PIERDERILOR DE SARCINA LOCALE PE TRONSONUL J
C     F(J)-COEFICIENTUL DE REZISTENTA HIDRAULICA LINIARA AL TRONSONULUI J
C     DI(J)-DIAMETRUL INTERIOR AL CONDUCTEI,IN M
C     DELC(J)-GROSIMEA FERETELUI CONDUCTEI J,IN M
C     V(J)-VITEZA REALA A AGENTULUI TERMIC DIN TRONSONUL J,IN M/S
C     R(J)-REZISTENTA TERMICA TOTALA A TRONSONULUI J LA TRANSFERUL
C          CALDURII,IN M.GRD/W
C     DSO(K)-LISTA DIAMETRELOR STANDARDIZATE DIN OTEL,IN M
C     DSC(K)-LISTA DIAMETRELOR STANDARDIZATE DIN CUPRU,IN M
C     DELSO(I)-GROSIMEA PERETELUI CONDUCTEI CORESPUNZATOARE DIAMETRULUI
C           STANDARDIZAT I DIN OTEL,IN M
C     DELSC(I)-GROSIMEA PERETELUI CONDUCTEI CORESPUNZATOARE DIAMETRULUI
C           STANDARDIZAT I DIN CUPRU,IN M
C     DELIZ(J)-GROSIMEA OPTIMALA A IZOLATIEI TRONSONULUI J,IN MM
C     FCMIN(J),FEMIN(J),FCEMIN(J)-VALOAREA MINIMA A FUNCTIEI CRITERIALE
```


Scanned with OKEN Scanner

```
C            DE OPTIMIZAT RESPECTIV DUPA CRITERIUL ECONOMIC.ENERGETIC
C            SI ECONOMICO-ENERGETIC
C     MC-IDENTIFICATOR AVIND VALAREA 0 SAU 1 DUPA CUM MATERIALUL
C       CONDUCTELOR ESTE OTELUL,RESPECTIV CUPRUL
C     KOD-IDENTIFICATOR CU VALOAREA 0 SAU 1 DUPA CUM SE FOLOSESTE LA
C        CALCULUL IZOLATIEI METODA OPTIMIZARII SAU METODA CLASICA
C     KIP-IDENTIFICATOR CU VALOAREA 0 SAU 1 DUPA CUM SE EFECTUEAZA
C        CALCULUL HIDRAULIC SI TERMIC SAU NUMAI CALCULUL TERMIC AL
C        CONDUCTELOR FRIGORIFICE
C
C    ALOCAREA MEMORIEI
      DIMENSION G(150),VI(150),V(150),ZITA(150),F(150),DI(150),H(150),
     *TE(150),TI(150),TPR(150),A(150),RO(150),RC(150),RE(150),R(150),
     *DELIZC(150),DELIZE(150),DELIZ(150),FCMIN(150),FEMIN(150),FCEMIN(150),
     *FC(50),FE(50),FCE(50),DELC(150),W(150),DSO(40),DELSO(40),DSC(15),
     *DELSC(15)
      REAL LAMC,LAMP,LAMIZ,LAMBDA(150),NIU(150),L(150)
C
C    INTRODUCEREA DATELOR SI SCRIEREA LOR
      DATA DSC/0.0045,0.005,0.0065,0.007,0.009,0.0105,0.011,0.0125,
     *0.013,0.015,0.017,0.019,0.020,0.0235,0.0325,/,DELSC/1.5,
     *1.0,1.5,1.0,1.0,1.5,1.0,1.5,1.0,1.0,1.0,1.0,1.0,1.5,2.5/,
     *DSO/0.009,0.01,0.014,0.015,0.016,0.019,0.020,0.021,0.025,
     *0.026,0.027,0.033,0.034,0.035,0.041,0.042,0.050,0.051,0.066,
     *0.067,0.068,0.069,0.079,0.080,0.081,0.098,0.099,0.100,
     *0.123,0.124,0.125,0.147,0.148,0.203,0.227,0.229,0.255,
     *0.304,0.306,0.357/,DELSC/0.0025,0.0020,0.0030,0.0025,0.0020,
     *0.0030,0.0025,0.0020,0.0035,0.0030,0.0035,0.0035,0.0030,0.0025,
     *0.0035,0.0030,0.0035,0.0030,0.0050,0.0045,0.0045,0.0035,0.0050,
     *0.0045,0.004,0.005,0.0045,0.040,0.050,0.045,0.040,0.006,0.0055,
     *0.008,0.009,0.008,0.009,0.010,0.009,0.010/
      READ (1,1) NC,MC,LAMC,DELP,LAMP,B0,B,EPS,KOD,KIP
    1 FORMAT (2I5,4F10.3,F10.6,F5.3,2I5)
      IF (KOD.NE.0) GOTO 3
      READ (1,98) EIZ,TAU,T,SIGMA,PSI
    2 FORMAT (F6.2,3F5.2)
    3 IF (KIP.EQ.0) GOTO 4
      DO 94 J=1,NC
      READ (1,73) DI(J),DELC(J),G(J),T(J),TI(J),TPR(J),LAMBDA(J),
     *NIU(J),A(J),W(J)
   73 FORMAT (2F10.4,F10.7,F5.1,F6.1,F5.1,3F6.3,F6.1)
   94 CONTINUE
      GOTO 74
    4 DO 6 J=1,NC
      READ (1,5) L(J),G(J),ZITA(J),VI(J),TE(J),TI(J),TPR(J),LAMBDA(J),
     *NIU(J),A(J),RO(J),W(J)
    5 FORMAT (F5.1,F10.7,3F5.1,F6.1,F5.1,3F6.3,F8.2,F5.1)
    6 CONTINUE
   74 WRITE (3,7)
    7 FORMAT (////,22X,36('*'),/,22X,'*P R O G R A M U L  C O F R I Z O P*',
     */,22X,36('*'))
      IF (KIP.NE.0) GOTO 75
      WRITE (3,90)
   90 FORMAT (//,11X,'CALCULUL HIDRAULIC AL CONDUCTELOR FRIGORIFICE SI',
     */,11X,48('-'),/,17X,'DIMENSIONAREA IZOLATIEI ACESTORA',/,17X,32('-'))
      GOTO 76
   75 WRITE (3,77)
   77 FORMAT (//,11X,'DIMENSIONAREA IZOLATIEI CONDUCTELOR FRIGORIFICE',
     */,11X,47('-'))
   76 WRITE (3,8)
    8 FORMAT (///,17X,'VERIFICAREA PARAMETRILOR DE INTRARE',/,17X,35('='),
     *///,27X,'DATE GENERALE',/,27X,13('*'))
      WRITE (3,9) MC,NC,LAMC,DELP,LAMP,B0,B,EPS,KOD,KIP
    9 FORMAT (/,28X,'MC=',I2,3X,'NC=',I2,//,9X,'LAMC=',F5.1,' W/(M.GRD)',
     *2X,'DELP=',F5.3,' M',2X,'LAMP=',F5.3,' W/(M.GRD)'.//,13X,'B0=',F5.3,
     *2X,'B=',F7.5,2X,'EPS=',F5.3,' M',2X,'KOD=',I1,2X,'KIP=',I1)
      IF (KOD.NE.0) GOTO 11
      WRITE (3,97) EIZ,TAU,T,SIGMA,PSI
   97 FORMAT (/,1X,'EIZ=',F6.1,' KWH/MC',2X,'TAU=',F6.1,' ORE/AN',2X,
     *'T=',F4.1,' ANI',2X,'SIGMA=',F4.2,2X,'PSI=',F4.2)
      IF (PSI.EQ.0) GOTO 11
      WRITE (3,10) CIZ,P,A0,CF
   10 FORMAT (/,9X,'CIZ=',F7.1,' LEI/MC',3 X,'P=',F4.2,3X,'A0=',F4.2,3X,
     *'CF=',F4.2,' LEI/KWH')
   11 WRITE (3,14)
   14 FORMAT (///.14X,'CARACTERISTICILE TRONSOANELOR SI TEMPERATURILE',
     */,14X,46('*'))
      IF (KIP.NE.0) GOTO 79
      WRITE (3,76)
   76 FORMAT (//,1X,97('-'),/,1X,'I',2X,'J',3X,'L(J). M',2X,'G(J), MC/S'.
     *2X,'ZITA(J)',2X,'VI(J), M/S',2X,'TE(J), GRD',2X,'TI(J), GRD',3X,
     *'TPR(J)'.6X,'W(J). M/S',1X,'I',/,1X,'I',95('-'),'I')
      DO 16 J=1,NC
      WRITE (3,15) J,L(J),G(J),ZITA(J),VI(J),TE(J),TI(J),TPR(J),W(J)
   15 FORMAT (1X,'I',95X,'I',/,1X,'I',I3,3X,F5.1,2X,F10.7,4X,F5.1.6X.F4.1,
     *6(7X,F5.1),5X,'I')
      IF (J.NE.NC) GOTO 16
      WRITE (3,55)
   55 FORMAT (1X,97('-'))
   16 CONTINUE
      GOTO 104
   79 WRITE (3,87)
   87 FORMAT (//.1X,87('-'),/,1X,'I',2X.'J',2X,'DI(J), M',2X.'G(J). MC/S'.
     *1X,'DELC(J), M'.2X,'TE(J), GRD'.2X,'TI(J), GRD'.3X,'TPR(J)',6X,
     *'W(J), M/S'.1X,'I',/,1X.'I'.85('-'),'I')
      DO 103 J=1.NC
      WRITE (3.88) J.DI(J),G(J).DELC(J).TE(J).TI(J).TPR(J).W(J)
   88 FORMAT (1X,'I',85X,'I'./,1X,'I',I3,2X,F6.4,2X F10.7,5X,F6.4,5X,
     *F5.1.3(7X,F5.1),5X,'I')
      IF (J.NE.NC) GOTO 103
      WRITE (3,89)
   89 FORMAT (1X,87,('-'))
  103 CONTINUE
  104 WRITE (3,17)
   17 FORMAT (///,16X,'CARACTERISTICILE AGENTULUI FRIGORIFIC',/,16X,
     *37('*'),//,2X,67('-'),/,2X,'I',2X,'J',3X,'LAMBDA(J), W/(M.GRD)',
     *2X,'NIU(J), MP/S',2X,'A(J), MP/S',2X,'RO(J), KG/MC',1X,'I',/,2X,
     *'I',65('-'),'I')
      DO 19 J=1,NC
      IF (KIP.NE.0) GOTO 107
      WRITE (3,18) J.LAMBDA(J),NIU(J),A(J),RO(J)
   18 FORMAT (2X,'I',65X,'I',/,2X,'I',I3,9X,F6.3,13X,F4.2,9X.F4.2,
     *8X,F6.1,3X,'I')
      GOTO 109
  107 WRITE (3,108) J,LAMBDA(J),NIU(J),A(J)
  108 FORMAT (2X,'I' 65X,'I',/,2X,'I',I3,9X,F6.3,13X,F4.2,9X.F4.2,
     *10X,'-',6X,'I')
  109 IF (J.NE.NC) GOTO 19
      WRITE (3,56)
   56 FORMAT (2X,67(')'))
   19 CONTINUE
C
C    EFECTUAREA CALCULELOR SI SCRIEREA REZULTATELOR
      WRITE (3,20)
   20 FORMAT (///,29X,'REZULTATELE CALCULULUI',/,29 X,22('='))
      IF (KIP.NE.0) GOTO 105
      WRITE (3,21)
   21 FORMAT (//,28X,'DIMENSIONAREA CONDUCTELOR',/,28X,25('*'),
     *//,2X,102('-'),/,2X,'I',1X,'J',3X,'L(J), M',2X,'TI(J), GRD',2X,
     *'RO(J), KG/MC',2X,'DI(J), M',2X,'DELC(J), M',2X,'V(J), M/S',2X,
     *'ZITA(J)',3X,'F(J)',3X,'H(J), N/MP','I',/,2X,'I',100('_'),'I')
      DO 25 J=1,NC
      DI(J)=SQRT(1.273*G(J)/VI(J))
      K=1
      IF (MC.EQ.0) GOTO 70
   22 IF (DI(J).LE.DSC(K)) GOTO 23
      K=K+1
      GOTO 22
   70 IF (DI(J).LE.DSO(K)) GOTO 71
      K=K+1
      GOTO 70
   23 DI(J)=DSC(K)
      DELC(J)=DELSC(K)/1000.
      GOTO 72
   71 DI(J)=DSO(K)
      DELC(J)=DELSO(K)
   72 V(J)=1.273*G(J)/DI(J)**2.
      REY=10.**6.*V(J)*DI(J)/NIU(J)
      F(J)=1./(1.8*ALOG10(REY)-1.5)**2.
      H(J)=(F(J)*L(J)/DI(J)+ZITA(J))*V(J)**2.*RO(J)/2.
      WRITE (3,24) J,L(J),TI(J),RO(J),DI(J),DELC(J),V(J),ZITA(J),F(J),H(J)
   24 FORMAT (2X,'I',100X,'I',/,2X,'I',I3,3X,F5.1,5X,F5.1,8X.F6.1,6X.F5.3,
     *6X,F6.4,6X,F4.1,6X,F5.1,3X,F6.4,2X,F10.1,'I')
      IF (J.NE.NC) GOTO 25
```


Scanned with OKEN Scanner

```
      WRITE (3,57)
   57 FORMAT (2X,102('-'))
   25 CONTINUE
      GOTO 110
  105 DO 106 J=1,NC
      V(J)=1.273*G(J)/DI(J)**2.
  106 CONTINUE
  110 DO 43 J=1,NC
      DE=DI(J)+2.*DELC(J)
      REY=10.**6.*V(J)*DI(J)/NIU(J)
      PR=NIU(J)/A(J)
      IF (TI(J).GE.TE(J)) GOTO 101
      C=0.4
      GOTO 102
  101 C=0.3
  102 ALFAI=LAMBDA(J)/DI(J)*0.023*REY**0.8*PR**C
      IF (W(J).EQ.0.) GOTO 26
      ALFAE=11.6++7.*SQRT(W(J))
      GOTO 91
   26 ALFAE=4.
   91 IF (TI(J).LT.TE(J)) GOTO 27
      LAMIZ=BO+B*(TI(J)+TE(J))/2.
      GOTO 92
   27 LAMIZ=BO+B*(TI(J)+TPR(J))/2.
   92 IF (KOD.NE.0) GOTO 80
      IF (PSI.EQ.0.) GOTO 33
      K=1
      DELIZC(J)=0.0
      IF (DE.LE.(2*LAMIZ/ALFAE)) GOTO 32
   28 CALL DIZPRO (J,DE,DELP,DELIZC,DIZ,DP)
      IF (TI(J).GE.TE(J)) GOTO 30
      C1=DIZ/DE*ALOG(DIZ/DE)
      C2=2.*LAMIZ/ALFAE/DE*((TE(J)-TI(J))/(TE(J)-TPR(J))-1.)
      IF (C1.GE.C2) GOTO 30
   29 DELIZC(J)=DELIZC(J)+EPS
      GOTO 28
   30 CALL REZTERM (J,ALFAI,ALFAE,DI,DE,DP,DIZ,LAMC,LAMP,LAMIZ,RC)
      S=((1.+AO)**T-1.)/(AO*(1.+AO)**T)
      FC(K)=(S*P+1.)*CIZ*DELIZC(J)*(DE+DELIZC(J))+S*SIGMA*TAU*CF*
     *ABS(TE(J)-TI(J))/(1000.*RC(J))
      IF (K.EQ.1.) GOTO 31
      IF (FC(K).LT.FC(K-1)) GOTO 31
      FCMIN(J)=FC(K)
      DELIZC(J)=DELIZC(J)*1000.
      GOTO 32
   31 K=K+1
      GOTO 29
   32 IF(PSI.EQ.1.) GOTO 43
   33 K=1
      DELIZE(J)=0.0
      IF (DE.LE.(2*LAMIZ/ALFAE)) GOTO 38
   34 CALL DIZPRO (J,DE,DELP,DELIZE,DIZ,DP)
      IF (TI(J).GE.TE(J)) GOTO 36
      C1=DIZ/DE*ALOG(DIZ/DE)
      C2=2.*LAMIZ/ALFAE/DE*((TE(J)-TI(J))/(TE(J)-TPR(J))-1.)
      IF (C1.GE.C2) GOTO 36
   35 DELIZE(J)=DELIZE(J)+EPS
      GOTO 34
   36 CALL REZTERM (J,ALFAI,ALFAE,DI,DE,DP,DIZ,LAMC,LAMP,LAMIZ,RE)
      FE(K)=EIZ*DELIZE(J)*(DE+DELIZE(J))+SIGMA*TAU*TE*ABS(TE(J)-TI(J))
     */(1000.*RE(J))
      IF (K.EQ.1) GOTO 37
      IF (FE(K).LT.FE(K-1)) GOTO  37
      FEMIN(J)=FE(K)
      DELIZE(J)=DELIZE(J)*1000.
      GOTO 38
   37 K=K+1
      GOTO 35
   38 IF (PSI.EQ.0.) GOTO 43
   80 K=1
      DELIZ(J)=0.0
      IF (DE.LE.(2*LAMIZ/ALFAE))GOTO 43
   39 CALL DIZPRO (J,DE,DELP,DELIZ,DIZ,DP)
      IF (TI(J).GE.TE(J)) GOTO 41
      C1=DIZ/DE*ALOG(DIZ/DE)
      C2=2.*LAMIZ/ALFAE/DE*((TE(J)-TI(J))/(TE(J)-TPR(J))-1.)
      IF (C1.GE.C2) GOTO 41
   40 DELIZ(J)=DELIZ(J)+EPS
      GOTO 39
   41 CALL REZTERM (J,ALFAI,ALFAE,DI,DE,DP,DIZ,LAMC,LAMP,LAMIZ,R)
      IF (KOD.NE.0.) GOTO 81
      S=((1.+AO)**T-1.)/(AO*(1.+AO)**T)
      F1=(S*P+1.)*CIZ*DELIZ(J)*(DE+DELIZ(J))/S*SIGMA*TAU*CF
     **ABS(TE(J)-TI(J))/(1000.*R(J))
      F2=EIZ*DELIZ(J)*(DE+DELIZ(J))+
      SIGMA*TAU*T*ABS(TE(J)-TI(J))/(1000.*R(J))
      FCE(K)=SQRT(PSI*ABS(F1-FCMIN(J))**2.+(1.-PSI)*ABS(F2-FEMIN(J))**2.)
      IF (K.EQ.1) GOTO 42
      IF (FCE(K).LT.FCE(K-1)) GOTO 42
      FCEMIN(J)=FCE(K)
      DELIZ(J)=DELIZ(J)*1000.
      GOTO 43
   42 K=K+1
      GOTO 40
   81 IF (TI(J),LT,TE(J)) GOTO 93
      IF (ABS(J)-TPR(J)),GT.0,05) GOTO 40
   93 DELIZ(J)=DELIZ(J)*1000.
   43 CONTINUE
      IF (KOD.NE.0) GOTO 85
      WRITE (3,44)
   44 FORMAT (//,15X,'CALCULUL OPTIMAL AL IZOLATIEI FRIGORIFICE',/,15X,
     *41('*'))
      IF (PSI,EQ,0,) GOTO 48
      WRITE (3,45)
   45 FORMAT (//,25X,'A) CRITERIUL ECONOMIC',/,28 X,18('-'),//,
     *13X,46('-'),,/,13X,'I',1X,'J',3X,'R(J), M.GRD/W',3X,'DELIZ(J), MM',
     *3X,'FCMIN(J)','I',/,13X,'I',44('-'),'I')
      DO 47 J=1,NC
      WRITE (3,46) J,RC(J),DELIZC(J),FCMIN(J)
   46 FORMAT (13X,'I',44X,'I',/,13X,'I',I3.4X,F8.3,8X,F5.1,6X,F10,3,'I'I
      IF (J.NE.NC) GOTO 47
      WRITE (3,58)
   58 FORMAT (13X,46('-'))
   47 CONTINUE
      IF (PSI,EQ,1,) GOTO 100
   48 WRITE (3,49)
   49 FORMAT (//,25X,'B) CRITERIUL ENERGETIC',/,26X,19('-'),//,
     *13X,46('-'),,/,13X,'I',1X,'J',3X,'R(J), M.GRD/W',3X.'DELIZ(J), MM',
     *3X,'FEMIN(J)','I',/,13X,'I',44('-'),'I')
      DO 50 J=1,NC
      WRITE (3,46) J,RE(J),DELIZE(J),FEMIN(J)
      IF (J.NE.NC) GOTO 50
      WRITE (3,58)
   50 CONTINUE
      IF (PSI.EQ.0.) GOTO 100
      WRITE (3,51)
   51 FORMAT (//,20X,'C) CRITERIUL ECONOMICO-ENERGETIC',/,23X,29('-'),//,
     *13X,46('-'),/,13X,'I',1X,'J',3X,'R(J), M.GRD/W',3X,'DELIZ(J), MM',
     *1X.'FCEMIN(J)','I',/,13X,'I',44('-'),'I')
      DO 52 J=1,NC
      WRITE (3,46) J,R(J),DELIZ(J),FCEMIN(J)
      IF (J.NE.NC) GOTO 52
      WRITE (3,58)
   52 CONTINUE
      GOTO 100
   85 WRITE (3,82)
   82 FORMAT (//,20X,'CALCULUL IZOLATIEI FRIGORIFICE',/,20X,30('*'),
     *//,17X,35('-'),/,17X,'I',1X.'J'.3X,'R(J), M.GRD/W',3X,
     *'DELIZ(J), MM','I',/.17X,'I',33('-'),'I')
      DO 86 J=1,NC
      DELIZ(J)=DELIZ(J)*1.2
      WRITE (3,83) J,R(J),DELIZ(J)
   83 FORMAT (17X,'I',33X,'I',/,17X,'I',13,4X,F8,3,8X,F5,1.5X,'I')
      IF (J.NE.NC) GOTO 86
      WRITE (3,84)
   84 FORMAT (17X,35('-'))
   86 CONTINUE
  100 STOP
      END
```


Scanned with OKEN Scanner

```
      SUBROUTINE DIZPRO (J,DE,DELP,DELIZ,DIZ,DP)
C     CALCULUL DIAMETRULUI IZOLATIEI SI AL PROTECTIEI ACESTEIA
      DIMENSION DELIZ(150)
      DIZ=DE+2.*DELIZ(J)
      DP=DIZ+2.*DELP
      RETURN
      END
```

```
      SUBROUTINE REZTERM (J,ALFAI,ALFAE,DI,DE,DP,DIZ,LAMC,LAMP,LAMIZ,R)
C     CALCULUL REZISTENTEI TERMICE TOTALE
      DIMENSION DI(150),R(150)
      REAL LAMIZ
      R1=1,/(3,14*ALFAI*DI(J))
      R2=1,/(3,14*ALFAE*DP)
      R3=1,/LAMC*ALOG(DE/DI(J))
      R4=1,/LAMIZ*ALOG(DIZ/DE)
      R5=1,/LAMP*ALOG(DP/DIZ)
      R(J)=R1+R2+1,/6,28*(R3+R4+R5)
      RETURN
      END
```

BIBLIOTECA ... "GH ASACHI" IAȘI

Scanned with OKEN Scanner

Bibliografie

1. ANDREI, N. RĂSTURNOIU, G. *Matrici rare și aplicațiile lor*, Editura Tehnică, București, 1983.
2. BÂRSAN, E. IGNAT, C. *Analiza rețelelor de distribuție a apei folosind „modelul conținut"*, Rev. Hidrotehnica, nr. 8, 1986.
3. BELLMAN, R. DREYFUS, S. *Programarea dinamică*, Editura Tehnică, București, 1967.
4. BERGE, C. *Théorie des graphes et ses applications*, Dunod, Paris, 1968.
5. BERTSEKAS, D.P. *Constrained optimization and Lagrange multiplier methods*, Academic Press, New York, 1982.
6. BOGOSLOVSKI, P.A. *Ledvii regim truboprovodov ghidroelektriceskih stanții*, Gosenergoizdat, Moskva, 1950.
7. BOROȘ, E. OPRIȘ, D. *Introducere în optimizarea liniară*, Editura Facla, Timișoara, 1979.
8. BRĂTIANU, C. *Calculul distribuției de temperaturi în izolația conductelor termice*, Rev. Construcțiilor de mașini, nr. 10, 1986.
9. BRĂTIANU, C. VORONCA, M. *Program de calcul cu elemente finite pentru determinarea cîmpurilor termice conductive neliniare bidimensionale*. Manual de utilizare, I.P. București, 1987.
10. BRĂTIANU, C. *Metode cu elemente finite în dinamica fluidelor*, Editura Academiei, București, 1983.
11. BRENT, R. P. *Algorithms for minimization without derivatives*, Prentice-Hall, London 1973.
12. BUCUR, C. *Metode numerice*, Editura Facla, Timișoara, 1983.
13. BURDUCEA, C. LECA, A. *Conducte și rețele termice*, Editura Tehnică, București, 1974.
14. CARABOGDAN, GH. ș.a. *Instalații termice industriale*, Editura Tehnică, București, 1983.
15. CARLIER, M. *Hydraulique générale et appliquée*, Eyrolles, Paris, 1980.
16. CÉA. J. Optimisation-*Théorie et algorithmes, Dunod*, Paris, 1971.
17. CHUNG, T.J. *Finite element analysis in fluid dynamics*, Mc Graw-Hill, New-York, 1978.
18. CREȚU, GH. *Optimizarea sistemelor de gospodărire a apelor*, Editura Facla, Timișoara, 1980.
19. CUTEANU, E. MARINOV, R. *Metoda elementelor finite în proiectarea structurilor*, Editura Facla, Timișoara, 1980.
20. DANCEA, I. *Programarea calculatoarelor numerice*, Editura Dacia, Cluj-Napoca, 1973.
21. DANCEA, I. *Metode de optimizare*, Editura Dacia, Cluj-Napoca, 1976.
22. DAVIS, P.J. RABINOWITZ, P. *Methods of numerical integration*, Academic Press, New York, 1975.
23. DÉMIDOVITCH, B. MARON, I. *Éléments de calcul numérique*, Mir, Moscou, 1979.
24. DINICĂ, G. *Metode variaționale și aplicații*, Editura Tehnică, București, 1980.
25. DODESCU, GH. ș.a., *Calculatoare electronice și sisteme de operare*, Editura Didactică și Pedagogică, București, 1975.
26. DODESCU, GH. *Metode numerice în algebră*, Editura Tehnică, București, 1979.
27. DORN, W. CRACKEN, D. *Metode numerice cu programe în Fortran*, Editura Tehnică, București, 1976.
28. DRAGOMIRESCU, M. MALIȚA, M. *Programare neliniară*, Editura Științifică și Enciclopedică, București, 1972.

Scanned with OKEN Scanner

29. DUMITRESCU, I. *Aplicații inginerești ale calculatoarelor*, Editura Didactică și Pedagogică, București, 1976.
30. DUMITRESCU, L. *Instalații sanitare pentru ansambluri de clădiri*, Editura Tehnică, București, 1980.
31. DUMITRU, V. *Programare neliniară*, Editura Academiei, București, 1975.
32. FIACCO, A.V. CORMICK, C.P. *Nonlinear programming*, John Wiley, New-York, 1968.
33. FLETCHER, R. *Optimization*, Academic Press, London, 1969.
34. FLEURY, C. *Primal and dual methods in structural optimisation*, Journal of the structural Division, nr. 5, 1980.
35. FORSYTHE, G.E. WASORW, W.R. *Finite difference methods for partial differential equation*, John Wiley, New York, 1960.
36. FRIEDRICH, R. *Bazele metodei elementelor finite*, Litografia I.P. Timișoara, 1972.
37. GASS, S.I. *Linear programming*, Mc Graw-Hill, New York, 1969.
38. GÂRBEA, D. *Analiză cu elemente finite*, Editura Tehnică, București, 1990.
39. GHEORGHE, G. *Distribuția și utilizarea gazelor naturale*, Editura Tehnică, București, 1972.
40. GILL, P. ș.a. *Practical optimization*, Academic Press, New York, 1981.
41. GODUNOV, S.K. REABENKI, V.S. *Scheme de calcul cu diferențe finite*, Editura Tehnică, București, 1977.
42. HENRICI, P. *Discrete variable methods for ordinary differential equations*, John Wiley, New York, 1962.
43. HIMMELBLAN, D.M. *Applied nonlinear programming*, Mc Graw-Hill, New York, 1972.
44. HINTON, E. OWEN, D. *Finite element programming*, Academic Press, New York, 1977.
45. IAMANDI, C. ș.a. *Hidraulica instalațiilor*, Editura Tehnică, București, 1985.
46. ILICI, C. *Probleme de optimizare și algoritmi de aproximare a soluțiilor*, Editura Academiei, București, 1980.
47. ION, M. *Calculul structurilor hidrotehnice*, Litografia I.P. Timișoara, 1982.
48. IONESCU, V. POPEEA, C. *Optimizarea sistemelor*, Editura Didactică și Pedagogică, București, 1981.
49. IONESCU, T. *Grafuri*, Editura Didactică și Pedagogică, București, 1974.
50. IORGULESCU, I. *Metode de calcul a rețelelor de conducte folosind formulări variaționale*, Rev. Hidrotehnica, nr. 5, 1987.
51. IRONS, B.M. AHMAD, S. *Techniques of finite elements*, John Wiley, New York. 1980.
52. JURA, C. *Alimentări cu apă*, Litografia I.P. Timișoara, 1984.
53. KAUFMANN, A. *Metode și modele ale cercetării operaționale*. Editura Științifică și Enciclopedică, București, 1967.
54. KILYENI, ȘT. *Optimizări și calculatoare în electroenergetică*, Litografia I.P. Timișoara, 1987.
55. LECA, A. ș.a. *Conducte pentru agenți termici*, Editura Tehnică, București, 1986.
56. LEGRAS, J. *Methodes et techniques de l'analyse numérique*, Dunod, Paris, 1971.
57. LEONTE, A. VRACIU, G. *Elemente de calcul matriceal cu aplicații*, Editura Tehnică, București. 1975.
58. LITEANU, C. RÎCĂ, I. *Optimizarea proceselor analitice*, Editura Academiei, București, 1985.
59. MARINESCU, GH. ș.a. *Probleme de analiză numerică rezolvate cu calculatorul*, Editura Academiei, București, 1987.
60. MARUSCIAC, I. *Metode de rezolvare a problemelor de programare neliniară*, Editura Dacia, Cluj-Napoca, 1973.
61. MARCHOUK, G.I. *Méthodes de calcul numérique*, Mir, Moscou, 1980.
62. MĂRUȘTER, ȘT. *Metode numerice în rezolvarea ecuațiilor neliniare*. Editura Tehnică, București, 1981.
63. MORARU, F. ATODIROAEI, M. *Programarea microcalculatoarelor în sistemul de operare CP/M*, Editura Științifică și Enciclopedică, București, 1989.
64. NORRIE, D.H. VRIES, G. *An introduction to finite element analysis*, Academic Press, New York, 1978.
65. OLARIU, V. BRĂTIANU, C. *Modelarea numerică cu elemente finite*, Editura Tehnică, București, 1986.
66. PASCARIU, I. *Elemente finite*, Editura Militară, București, 1985.
67. PĂVĂLOIU, I. *Rezolvarea ecuațiilor prin interpolare*, Editura Dacia, Cluj-Napoca, 1981.
68. PCHÉNITCHNY, B. DANILINE, Y. *Méthodes numériques dans les problèmes d'extremum*, Mir, Moscou, 1977.
69. PICARD, C.F. *Graphes et questionnaires*, Gauthier-Wilars, Paris, 1971.
70. POPESCU, H. CHIROIU, V. *Calculul structurilor optimale*, Editura Academiei, București, 1981.

Scanned with OKEN Scanner

71. PUNN, L. *Introduction to optimization practice*, John Wiley, New York, 1969.
72. PURCARU, I. *Elemente de algebră şi programare liniară*, Editura Ştiinţifică şi Enciclopedică, Bucureşti, 1982.
73. RAFIROIU, M. *Modele ale cercetării operaţionale aplicate în construcţii*, Editura Tehnică, Bucureşti, 1980.
74. RAO, S. *The finite element method in engineering*, Pergamon Press. New York, 1981.
75. RAMSAY, J.O. *A family of gradient methods for optimization*, The Computer Journal, nr. 4, 1970.
76. RESA, I. ş.a. *Probleme de statistică rezolvate pe calculator*, Editura Facla, Timişoara, 1984.
77. RICE, J.R. *Approximation des functions*, Dunod, Paris, 1969.
78. SALVADORI, M.G. BARON, M.L. *Metode numerice în tehnică*, Editura Tehnică, Bucureşti, 1972.
79. SĂVULESCU, T. *Instalaţii de încălzire şi ventilare*, Editura Tehnică, Bucureşti, 1985.
80. SÂRBU, I. RETEZAN A. *Détermination du tracé optimal d'une magistrale d'adduction á l'aide de la thèorie des graphes*, Buletinul I.P. Timişoara, 1984.
81. SÂRBU, I. TOBOLCEA, V. *Optimierung des platzes einer fassung und der linie einer litungsmagistrale mit hilfe der dynamischen programmierung*, Buletinul I.P. Iaşi, 1985.
82. SÂRBU, I. *Asupra calculului hidraulic optim al canalelor cu taluze plane şi radier cilindric*, Rev. Hidrotehnica, nr. 2, 1986.
83. SÂRBU, I. *Optimisation du tracé de réseaux rammifiés des distribution d'eau*, Buletinul I.P. Timişoara, 1986.
84. SÂRBU, I. *Optimizarea dimensionării reţelelor de distribuţie a apei*, Rev. Hidrotehnica, nr. 1, 1986.
85. SÂRBU, I. *Model de calcul al regimului hidraulic în reţele complexe de distribuţie a apei*, Rev. Hidrotehnica, nr. 8, 1987.
86. SÂRBU, I. *Aspects de calcul de réseaux annulaires de distribution de gaz á travers la theorie des graphes*, Buletinul I.P. Timişoara, 1988.
87. SÂRBU, I. FEHER, A. *Model matematic şi program pentru dimensionarea optimală a reţelelor interioare de alimentare cu apă în clădiri*, Rev. Tehnic 2000, nr. 4, 1988.
88. SÂRBU, I. *Model matematic pentru studiul fenomenului de îngheţ al apei în conductele aeriene sub presiune*, Rev. Construcţii, nr. 6/7, 1988.
89. SÂRBU, I. *Metode şi procedee de echilibrare hidraulică a reţelelor inelare de distribuţie a apei*, Referat doctorat, nr. 2, U.T. Timişoara, 1990.
90. SÂRBU, I. *Instalaţii frigorifice* Litografia U.T. Timişoara, 1992.
91. SÂRBU, I. *L'optimisation de la répartition des débits dans les rêseaux annulaires de distribution d'eau*, Buletinul U.T. Timişoara, 1991.
92. SÂRBU, I. *Optimizarea dimensionării izolaţiilor frigorifice folosind calculatoarele electronice*, Buletinul simpozionului „Utilizarea calculatoarelor în domeniul învăţământului". Constanţa, 1991.
93. SÂRBU, I. *Model neliniar de optimizare a dimensionării reţelelor inelare complexe pentru alimentări cu apă*, Rev. Construcţii, 1991.
94. SÂRBU, I. *Metode numerice şi de optimizare în calculul instalaţiilor*, Litografia U.T. Timişoara, 1993.
95. SIMA, V. VARGA, A. *Practica optimizării asistată de calculator*, Editura Tehnică, Bucureşti, 1986.
96. STARANG, G. *The finit element method and approximation theory numerical solution of partial differential equation*, Academic Press, New York, 1970.
97. STĂNCESCU, I. *Bazele tehnice şi economice ale termoficării*, Editura Tehnică, Bucureşti, 1967.
98. STÖER, J. *Einführung in die numerische mathematik*, Springer-Werlag, Berlin, 1972.
99. STOIAN, V. ş.a. *Programe BASIC pentru inginerii constructori*, Litografia I.P. Timişoara, 1988.
100. SANDRU, A. *Prezentarea sistemului de operare MS-DOS*, Editura Cristian, Bucureşti, 1990.
101. ŞCHIOP, A. *Analiza unor modele de discretizare*, Editura Academiei, Bucureşti, 1978.
102. ŞTEFĂNESCU, A. ZIDĂROIU, C. *Cercetări operaţionale*, Editura Didactică şi Pedagogică, Bucureşti, 1981.
103. ŞTEFĂNESCU, D. ş.a. *Transfer de căldură şi masă*, Editura Didactică şi Pedagogică, Bucureşti, 1983.
104. TENTER, F.G. *Proiektirovanie tplovoi izolaţii electrostanţii i teplovîh setei*, Energhia, Leningrad, 1972.
105. TEODORESCU, N. *Introducere în fizica matematică*, Editura Tehnică, Bucureşti, 1970.
106. TOMA, M. OGĂDESCU, I. *Metode numerice şi subrutine*, Editura Tehnică, Bucureşti, 1980.

Scanned with OKEN Scanner

107. TOMESCU, I. *Grafuri și programare liniară*, Editura Didactică și Pedagogică, București, 1975.

108. TOPAL, N. *Contribuții privind procesul de absorbție din instalațiile frigorifice*, Rev. Construcții, nr. 1/2, 1989.

109. TROPPER, A.M. *An introduction to linear algebra*, Elsevier, New York, 1969.

110. VINTILĂ, ȘT. ș.a. *Instalații sanitare și de gaze* — îndrumar de proiectare, Editura Tehnică, București, 1987.

111. VRACIU, G. POPA, A. *Metode numerice cu aplicații în tehnica de calcul*, Editura Scrisul românesc, Craiova, 1982.

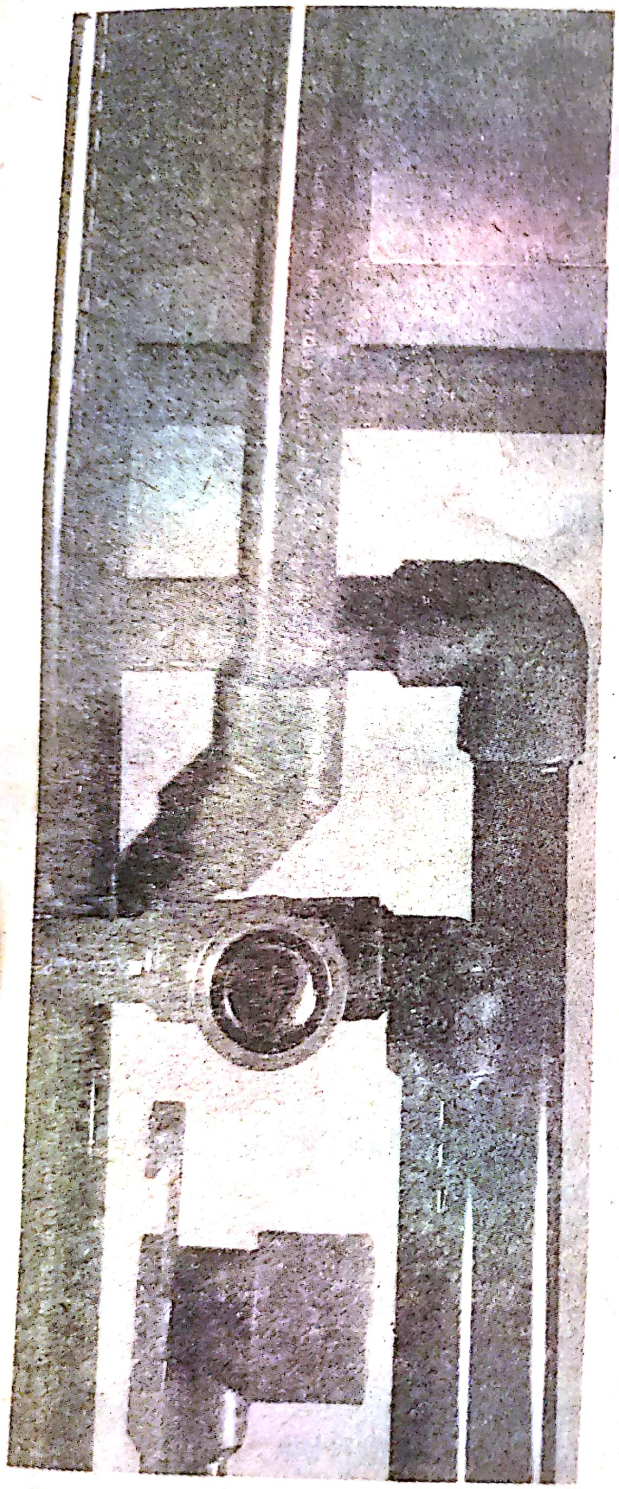
112. WALSH, G.R. *Methods of optimization*, John Wiley, London, 1975.

113. WEGGEL, L. *On a discrete version of the Newton-Raphson method*, SIAM, nr. 3, 1966.

114. *** Lucrările simpozionului „Instalațiile pentru construcții și confortul ambiental", Timișoara, 1992.

115. *** Lucrările colocviului tehnico-științific „Instalațiile pentru construcții și economia de energie", Iași, 1992.

Scanned with OKEN Scanner



S.C. ROMTEST S.R.L. TIMIŞOARA

Vă oferă:

CONDUCTE ŞI FITINGURI

Material:
- polipropilena

Domeniu de utilizare:
- instalaţii sanitare în clădiri noi sau vechi

Tehnica de montaj:
- sudare prin polifuziune

Caracteristici, avantaje:
- gama diametrelor 16...90 mm;
- presiunea de funcţionare de 10 bar;
- temperatura de durată de 60 °C;
- durata minimă de viaţă 50 ani;
- conductibilitatea termică scăzută (0,24 W/mK);
- rezistenţă mare la coroziune;
- pierderi de presiune mici;
- uşor adaptabile la instalaţiile existente.

FILTRE DE TRATARE A APEI

Scanned with OKEN Scanner

VENTILATOARE

UTILIZARE: aerisirea incintelor mici şi mijlocii (bucătării, băi, wc-uri, magazii şi magazine) precum şi industriale, fiind cuplate la canale de ventilaţie rigide sau flexibile.

AVANTAJE: nu pun probleme deosebite la proiectare, montaj şi exploatare; fiabilitate sporită în funcţionare; randament ridicat; capacitate de transport de 45...100000 mc/h.

SISTEME DE CLIMATIZARE

UTILIZARE: încălzirea, răcirea, filtrarea şi schimbarea aerului din incinte, folosindu-se aer exterior recirculat şi de amestec.

AVANTAJE: încălzire uşoară; înlocuiesc sursele convenţionale de încălzire.

CAZANE DE ÎNCĂLZIRE

UTILIZARE: centrale de încălzire proprie, folosind combustibil convenţional: gaz sau petrol.

AVANTAJE: randament până la 95%; durata de viaţă mare; montare uşoară cu ajutorul îmbinărilor prin presare.

Firma ROMTEST având colaborări cu diverse firme din străinătate (DEC, WOLF, KAMLEITHNER, HARREITHER, LIMODOR), va pune la dispoziţie pliante şi prospecte pentru consultare şi eventuale propuneri de contracte.

S.C.ROMTEST S.R.L. TIMIŞOARA
Str. D.Paciurea, nr.2 A
Tel/fax 096-133458.

Scanned with OKEN Scanner

LEI: 5.500

Scanned with OKEN Scanner

ISBN 973-31-0525-2